A Model of Owner Controlled, Full-Provenance, Non-Persistent, High-Availability Information Sharing

Sean Peisert
Berkeley Lab, CENIC, & UC Davis
California, USA
sppeisert@lbl.gov

Matt Bishop
UC Davis
California, USA
mabishop@ucdavis.edu

Ed Talbot
UC Davis
California, USA
edward.talbot@gmail.com

ABSTRACT

In this paper, we propose principles of information control and sharing that support ORCON (ORiginator COntrolled access control) models while simultaneously improving components of confidentiality, availability, and integrity needed to inherently support, when needed, responsibility to share policies, rapid information dissemination, data provenance, and data redaction. This new paradigm of providing unfettered and unimpeded access to information by authorized users, while at the same time, making access by unauthorized users impossible, contrasts with historical approaches to information sharing that have focused on need to know rather than need to (or responsibility to) share.

CCS CONCEPTS

• **Security and privacy → Formal security models**; **Systems security**; **Database and storage security**;

KEYWORDS

Access control, fault tolerance, information sharing, ORCON, provable security

ACM Reference Format:
Sean Peisert, Matt Bishop, and Ed Talbot. 2017. A Model of Owner Controlled, Full-Provenance, Non-Persistent, High-Availability Information Sharing. In *Proceedings of New Security Paradigms Workshop, California, USA, Oct. 2–4, 2017 (NSPW'17),* 11 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Some communities consider *need to know* and *originator controlled access control* (ORCON) [8] to be standard practice for information sharing and control. Need-to-know policies are intended to restrict the sharing of information to only the people who must have it. ORCON, which "...bases access on the creator of an object (or the information it contains)" [4, §4.5,§8.3], is designed to be a mechanism to enforce need-to-know policies.

However, institutional culture in a wide variety of communities is increasingly transforming from *need to know* to *need to share* and in some cases, a *responsibility to share* [7, 13, 21]. At the same time, ORCON must still be supported and perhaps even increased because as information is shared more widely, new mechanisms must be designed to protect it. Moreover, in many cases, such information must be provided securely yet in such a way that enables near-real-time use of the information.

Our objective for a *responsibility-to-share* model of information control and sharing, and the new paradigm that we present in this paper, is to provide unfettered and unimpeded access to information by authorized users, while at the same time, making access by unauthorized users impossible. Because historically, information sharing has focused on need to know rather than need to (or responsibility to) share, ORCON has failed to adequately support or even consider the needs of sharing rather than simply controlling.

1.1 Goals, Definitions, and Framework

In this paper, we discuss a variety of considerations that we believe can enhance *responsibility-to-share*-based ORCON. Several fundamental tenets underlie our solutions.

Only machines that ensure measurably and/or provably secure controls [16] can access (read or write) the data at issue. Data is highly replicated around the globe using Byzantine fault tolerance [9] mechanisms to maintain availability and integrity, and using cryptography and data slicing (e.g., *secret splitting* [20]) to maintain confidentiality. We also assume "always on," low latency high-bandwidth connections throughout this environment.

There are two types of entities who can "use" data: *originators* and *recipients*. Originators can directly read and write (which includes delete) data that they own. Any given data unit has exactly one originator at a time. Originators can specify the parameters for the treatment and behavior of data. For example, an originator can specify the conditions under which someone else can "take over" as the originator of the current data unit. One such condition might be if proof can be supplied that the originator has been incapacitated and as a result can no longer serve as an originator. In addition, an originator can specify other policies about its data, so if the originator dies, the data becomes available to all, to a specific group of entities, or is simply deleted. Originators can also grant access to entities ("recipients") to only read data. Recipients cannot write (or otherwise modify or delete) another originator's data. Originators and recipients must authenticate using measurably strong authentication [17].

Complete historical provenance information is maintained and bound to all data replicas currently available. The provenance information includes any accesses by any originator or recipient; all information about the replication of the data itself, such as where it is kept; any grant of access by an originator to a recipient; any transformation (such as encryption or decryption); any copying; and any destruction.

NSPW'17, Oct. 2–4, 2017, California, USA

The risk of disclosure — a breach of confidentiality — is only one consideration. Risks of (lack of) availability and integrity must also be considered, as must cost and performance. In some cases, the risk of disclosure cannot be pre-defined without a detailed understanding of the environment and context for information sharing, as well as an understanding of the requirements for performance and other attributes.

This calculation might also involve a variety of non-technical factors. For example, a digital solution to blocking the sharing of data in a file would prevent someone from mailing the file to another, but it would not prevent someone from taking pictures of the screen using an analog camera, developing the film, and mailing the print to someone. Acquiring large amounts of data this way would require substantially more cost and effort than acquiring small amounts of data because a large file will require many pictures, whereas mailing the file requires simply a set of commands. Rather than explicitly incorporate these factors into our model, the people implementing our model must identify which elements are the most applicable, and incorporate them.

We describe a spectrum of solutions from which one must be chosen based on cost, performance (processing, access time), and schedule. In order to best understand this spectrum, it is necessary to understand what the terms *owner* and *information* mean, and the ways in which information controls are needed under a variety of circumstances, such as when the most important threats are from external actors, and when to include internal actors in the threat analysis. As a side note, while in precise terms, we would typically define *data* as the representation of *information*, we have not made that distinction in this paper, and use the two words interchangeably.

2 EXISTING MODELS OF INFORMATION SHARING

The goals of the information control model that we describe in this paper encompass *both* the flow and the use of information. A variety of existing policy models have been developed, but those models do not meet our needs. For example, mechanisms that implement discretionary access control (DAC) do not distinguish between the owner of the file containing the data and the originator of the data, and hence are inadequate for addressing the needs of our model. Similarly, multilevel mandatory access control (MAC) models such as Bell-LaPadula [2] and Biba [3] were designed to model confidentiality and integrity in environments with rigidly defined levels of security and labels. But compartments or environments often have loose, "need to know" levels within levels. As such, these models place a considerable burden on each information sharer to serve as a declassification authority at every step of the process. Moreover, identifying discrete levels may be cumbersome if not impossible due to the "mosaic effect." This also rules out combinations of mechanisms involving multilevel MAC such as Lipner's integrity matrix model [10].

The ORCON model [8] is closest to meeting our goals. That model usually phrases requirements of sharing to a "need to know" basis. We generalize this. Thus, there may not be a "need to know" in order to share, but rather a "duty" to share. For example, a patient may not need to know which of two medical procedures the physician will use to treat a condition, but ethically (a non-technical consideration) the patient should be informed. Under traditional ORCON, the patient has no "need to know" because both treatments will achieve the same effect, so far as the patient is concerned. But certainly the physician has a duty to discuss the course of treatment with the patient, which our model could require. Further, we take into consideration the environment in which the sharing is to occur, as described in the next section. So, in some sense, our model expands some aspects of traditional ORCON models, and introduces constraints for other aspects of the model.

We note that the ORCON model does not address risk of "leaks" but is very much a "binary" model of sharing information (users either have access or not). For example, we do not address the issue of data intentionally or accidentally being shared with the "wrong" person. This is in large part because our motivation underlying the model is to help inform the design of future information sharing implementations for extremely security-sensitive systems, and thus build on a system that has an extremely tight focus on access as a design goal. Any aspect of "recovering" from an intentional or accidental leak is beyond the scope of this paper.

Our model is also influenced by the Traducement model of record security [22], which focuses on append-only integrity logs, as our approach requires for provenance tracking, but does not contain many of the other elements of our model.

The UCON model [14, 15], which extends ORCON, unifies access control, trust management, and digital rights management, bears surface level similarities to our approach, but focuses on the control of the rights for use rather than on information flow, as is our focus. UCON also assumes that information is in an encrypted digital container, an assumption not made here.

3 INFORMATION CONTROL DEGREES OF FREEDOM

Our model of information control identifies six key questions and considerations for how information can be treated:

Connected vs. Disconnected Access: can a recipient access information while in a "disconnected mode," or is information automatically destroyed, either immediately upon disconnection, or under other prescribed circumstances (e.g., a time duration) if the recipient disconnects from online communication?

Local Storage vs. Pointer Storage: can a recipient store information in their local store or are they limited to viewing a pointer to the information such that the information has no more presence on the recipient's computer than volatile storage in their video memory?

Ciphertext vs. Plaintext: can a recipient access an unencrypted copy of information or can he only access an encrypted copy, for example, processing it using homomorphic encryption? An encrypted file is of little value to an unauthorized user because that user lacks the key required for decryption. Therefore, even though the unauthorized user may be in possession of the information, the ability to access that information continues to be restricted. A user with the encryption key (ostensibly an authorized user) has unrestricted access to the information.

Replicated vs. Isolated Information: how widely is information replicated across diverse systems using diverse and independent paths such that availability and integrity of the information are both improved?

Sliced vs. Unified Information: when a particular set of information is at "rest," is it simply replicated to provide greater availability and integrity (potentially at the cost of confidentiality) or is it also sliced, so that without the ability to recall the individual packets and reassemble them in the proper order, an unauthorized user is incapable of reconstructing the original information.

Physical Controls vs. No Physical Controls: are physical controls in place to protect access to the information, or can the information be accessed from anywhere? Physical access to information can be limited through isolation. For example, gaining unauthorized access to a printed document secured in a vault requires (direct or indirect) physical access to the vault, which may be obtained by compromising the vault or someone who knows the combination to the vault. Similarly, digital information can be secured by placing the digital storage media in a physical vault or on an isolated system or network. Electronic access to digital media stored this way defeats physical protection by bypassing the safeguards provided by the vault.

These conditions all relate to methods of controlling access to information. Techniques may combine these conditions in varying ways and to varying degrees. For example, a submarine at sea requires access to information without network connections. But that environment may possess physical controls that make keeping a local, unencrypted store of information an acceptable risk. In contrast, a public cloud has no physical controls at the granularity of the data (they are at the granularity of the entire cloud), and so to protect information one might use a combination of replication and encryption and/or data slicing.

We now present axioms upon which our model is based.

Axiom 1. *Techniques exist to provide a means for ensuring data integrity and availability of data when that data is replicated at multiple locations.*

This axiom does not prescribe a method of ensuring this; it simply assumes such methods exist. An example of this is Byzantine fault tolerance (BFT). If the assumptions underlying BFT are met, then the axiom holds. If not, some other algorithm or method must be used, or the axiom will not hold in that environment.

Axiom 2. *Replicating the same copy of data in more places means there are more places that an attacker can corrupt that data.*

This axiom is the complement of Axiom 1. It states that if data is replicated in n places rather than $n + 1$ places, there is one less place for the attacker to violate the integrity of the data. It does *not* say that, in doing so, the attacker will corrupt all accesses to the data. Indeed, if the assumptions underlying BFT are met, then corrupting fewer than $n/3$ copies will not affect the integrity of the data that recipients see.

Axiom 3. *"Disconnected access" requires that a copy of the data be stored locally. Further, disconnected access and "pointer storage" are mutually exclusive.*

For our model, information is either local or remote. It is never both. Therefore, storing a reference to data (the "pointer") means that the data cannot be accessed locally; it must be accessed through the pointer. Note that pointer storage may enable the recipient to access a copy of the data stored on the local host, but if so then the recipient *must* make all accesses to that data using the pointer. She cannot access the data in any other way. The same is true for originators, of course.

Axiom 4. *Reading encrypted data of size n and key of size k units requires 2 accesses, costing $(n + k)a$, where a is the cost of accessing one unit.*

Axiom 5. *Reading data that has been split into s slices each of size d units requires at least s accesses, costing sda, where a is the cost of accessing one unit.*

These two axioms flow from the definitions of encryption and data slicing. Axiom 5 sets a lower bound, whereas Axiom 4 is a constant. Both axioms assume the cost of making an access, regardless of the amount of data sent or retrieved, is a. Also, the axioms deal only with fetching the data once. Modifying the data would, in most cases, require two accesses, one to view the data, and one to send the modifications.

It is important to note that, even though each slice is of size d, the assembled file is no larger than sd, because d will include meta-information describing (at least) the position of the slice in the data.

We now state four propositions.

Proposition 1. *"Encrypted access" and "data slicing" protect confidentiality by requiring multiple accesses to read the data.*

Proof. From Axiom 4, viewing encrypted data requires at least 2 accesses. From Axiom 5, viewing data that has been sliced requires at least s accesses, where s is the number of slices.. If $s = 1$, then the data has not been sliced, so $s \geq 2$. Therefore, both require multiple accesses. □

Proposition 2. *All other considerations being equal, encrypted data costs less to access when the size of the encryption key is less than the difference of the sliced data and the encrypted file.*

Proof. By Axiom 4, the cost of accessing encrypted data is $(n + k)a$; by Axiom 5, the cost of accessing sliced data is sda. Therefore, when $(n + k)a < dsa$, $k < ds - n$, proving the claim. □

Proposition 3. *Both encrypted access and data slicing raise the cost of accessing the data.*

Proof. Fetching a cryptographic key requires an access beyond retrieving the file, and fetching the slices require multiple accesses. The proposition follows immediately. □

Proposition 4. *"Byzantine state replication" and "data slicing" counter each other — more replication enhances availability and integrity while reducing confidentiality, and more slicing enhances confidentiality while reducing availability and integrity [16, §3.2].*

Proof. Consider Axioms 1 and 2. Let f be the number of faults to be tolerated. Let n be the number of replicas. Let r be the number of required slices (data dependent). Let a be the number of actual slices and s the number of actual state replicas. Thus, one data disclosure security metric DD with respect to machine replication is:

$$DD = \frac{n}{3f+1} \times \frac{a}{r}$$

Thus for each replica not "sliced," the system becomes *less* secure with regard to confidentiality, and for each additional slice, the machine becomes *more* secure with regard to confidentiality. By increasing both, then security with respect to all three primary goals can be increased without compromise. □

Using these propositions, we can explore a variety of ways to share information, including via combinations of encrypted access, data sharing, and Byzantine state replication. There is a partially-ordered spectrum of risk of disclosure associated with different degrees of "reads" that might be performed by a recipient that might be chosen based on need and tolerance for cost.

The lowest risk of disclosure is for a recipient to never even have a local copy of the data themselves but instead to use homomorphic encryption [6] (or some form of "simulated" homomorphic encryption [18]) to query remotely-stored encrypted copies of the data. Note that this data would be only accessible while the recipient is online and connected. If the recipient loses connectivity, the data would be instantly destroyed (e.g., using *Vanish* [5]). So this scheme prioritizes confidentiality above all else.

The highest risk of disclosure is for a recipient to be able to directly access an unencrypted, local copy of the data. For example, the data might be stored in video memory on the recipient's machine. In this case, the data should possess some set of conditions that would cause it to be destroyed (e.g., if a period of time is exceeded). Even in this case, if the recipient ever goes online after possessing a local copy, provenance information about which actions were (automatically) taken while the system was disconnected would be recorded along with a description of the "lost time story."

In between these two extremes are several methods for which the risk of lack of confidentiality and availability cannot be predefined without knowing more details about the context of use and/or storage—for example, the balance between encrypted access and data slicing, as described in Proposition 1, the counter-effects against each other of Byzantine state replication and data slicing as described in Proposition 4, and the costs of encrypted data and data slicing, under a variety of circumstances, as described in Propositions 2 and 3. One option is that a recipient has a local copy of the data but that data is encrypted and is never decrypted. That data is processed/analyzed by the recipient using homomorphic encryption. A second option is that a recipient does not have a local copy of the data but instead is able to query the remote data using homomorphic encryption.

The tradeoffs for these two options are the balance in risk of having a local, encrypted copy and accessing a remote, but unencrypted copy. The consideration of which poses more risk depends on how vulnerable the encrypted data is to decryption.Whenever the user actually possesses copies of the data, risk can be reduced by only providing the portion of the data required by the user and authorized by the originator. For example, a requester may require data from an originator for a formula in a specific cell of a table. The originator may grant access to only that piece of data. There is also a temporal aspect to access; the originator may also permit automatic updating of that data so any changes are reflected in the recipients table.

Risk also depends on purpose. Suppose Alice is the originator of Data Set A and Bob is an originator of Data Set B and the recipient of Data Set A. Suppose Bob requires a direct comparison of Data Set B to Data Set A. If Data Set B is of higher value, then then the lesser risk might be for Bob to compare the data sets locally on Bob's own machine using homomorphic encryption techniques. On the other hand, if Data Set A is of higher value, then the lesser risk might be for Bob to send his own Data Set B to Alice, who would usehomomorphic encryption on Alice's machine to perform the direct analysis.

This is close to a "transitivity property of information risk" (like the transitive property of equality) where the highest value data set determines the risk level (and therefore classification) of the overall data set. It does not, however, speak to the case where combining data sets of low value creates a high value data set simply through the association of the two data sets. This latter case is sometimes called the "mosaic effect."

However, as we have discussed, confidentiality is not the only metric for ORCON. Availability—the ability to obtain unfettered access under the proper conditions—must also be considered. For these reasons, we also pose the following goals to address the balance between confidentiality and availability, such as might arise when considering data slicing vs. Byzantine state replication in Proposition 3, but do so within an ORCON framework.

Goal 1. *Availability and integrity must be balanced with confidentiality, based on the particular scenario, as the properties can work against each other.*[1] *To maintain and maximize all three properties, additional resources must be expended.*

Goal 2. *Data should be associated with a provenance of all events occurring with that data, including viewing, modifying, replicating, computing, and redacting.*

Goal 3. *Data should be accessed only when the system accessing the data is connected to the network.*

Goal 4. *Data should be stored locally only temporarily and when in immediate use.*

We next state the transition rules for the model. For convenience, we list a set of preconditions for the rule.

Precondition 1. *Data is replicated widely in at least* $3f + 1$ *places, where* f *is the maximum number of replica failures that the system can tolerate before the entire system fails.*

Precondition 2. *Each data replica contains a provenance list identifying all authors who created or modified the data, the time of each modification, and the change made.*

[1] As indicated by Proposition 3, techniques to achieve availability and integrity and techniques to achieve confidentiality can counter each other.

Precondition 3. *Each data replica contains a provenance list identifying all viewers who viewed or (temporarily) copied the data and the time of viewing.*

Precondition 4. *Locally stored data is always encrypted, sliced, or protected with physical security.*

Creation Rule. When a user u creates data d, the data is stamped with its provenance information, which includes the creator u's identifier, no viewers as it has not been viewed, and is replicated on $3f + 1$ other machines. Data is always synchronized between connected replicas in real-time and data is synchronized with disconnected replicas when reconnection occurs. The data on those machines are protected using encryption or physical security measures or (if the data is sufficiently broken up) through data slicing.

Alteration Rule. When a user u alters or redacts data d, the appropriate provenance information, which includes the user identifier u and change c, are appended to the data's existing provenance list. This is replicated on $3f + 1$ other machines. Data is always synchronized between connected replicas in real-time and data is synchronized with disconnected replicas when reconnection occurs. The data on those machines are protected using encryption and/or physical security measures and/or (if the data is sufficiently broken up) through data slicing.

Computation Rule. When a user u performs a computation on data d, the data is first transferred to the machine(s) needed for the computation and stored as in the Creation Rule or the Alteration Rule using homomorphic encryption [6, 19],[2], secure multi-party computation [24], physical security measures, or (if the data is sufficiently broken up) through data slicing. Additionally, the act of obtaining and performing computation on the data, as well as other appropriate information such as the access time, is appended to the existing provenance list.

Connected Access Rule. A user u may access data d without physical access and physical security controls put in place to the system accessing the data only while connected to the network. The data d is accessed through "information pointers" and temporarily stored only in the local VRAM of u's machine.

Using a technique similar to capabilities in operating systems makes control of the data simpler. If the information pointer points directly to the information, then that information cannot be redacted in some documents and present in others. To enable this, rather than pointing directly to the information, the information pointer points to a second pointer in a repository under the control of the author(s) of the information. Then, to replace or delete the information, the second pointer is changed to point to the new information or a null area.

Disconnected Access Rule. If a user u wants "disconnected access" to data d then such access is only allowed when physical security is provided and the data contains a destruction "time bomb." Disconnected access to data d requires making the disconnected replica a full copy of data d. Additionally, the act of obtaining the data as well as individual viewings of the data, a unique user identifier, access time, and other appropriate information are appended to the existing provenance list. The data and provenance information is synchronized upon reconnection or destroyed after a pre-determined period of time.

Disconnected operation is sometimes a requirement: performance and access time for a submarine demand the ability to access data while offline. Disconnected access favors physical security in part because it rules out slicing and in part because an encryption key cannot be remotely transmitted to decrypt the information. But under the right circumstances, such as being on a submarine, physical security is sufficient to remove the burden of encryption and slicing.

These rules lead to three modes of accessing information.

(1) In the mode of *connected access for viewing*, data is replicated widely, the person accessing the data is always connected, and the data is accessed in such a way that it is stored only in the VRAM of that person's machine.
(2) In the mode of *connected access for processing*, the data is processed on a system in such a way that it is processed entirely in memory, or if it is stored on the system, it is stored using fully homomorphic encryption, stored using physical protection, or stored heavily distributed and sliced.
(3) In the mode of *disconnected access*, the person accessing the data is disconnected, so physical security coupled with measurably strong authentication and possible multi-person access rules provide the basis for securing the data. The data is still replicated and replicas re-sync whenever the connection is re-established. Further, the data in any single replica remaining disconnected longer than a pre-specified time t will be automatically destroyed and become unaccessible.

We note that if a system initially satisfies all four preconditions, then the system satisfies all four preconditions after any sequence of applications of the five rules. A proof for this is given in the Appendix. If one defines a system meeting all five rules as "secure," this theorem is analogous to the Basic Security Theorem of the Bell-LaPadula Model [2], in that it states systems beginning in a "secure" state and using these transition rules will always remain in a "secure" state.

A state flow diagram of this model, demonstrating the application of these rules, is shown in Figure 1.

4 INFORMATION OWNERSHIP

A major challenge in information sharing is determining the *owner* of the information. Moreover, it can also be challenging to even define *information*. Typically, even small documents consist of input from multiple authors. There may be an "author of record" who is responsible for releasing a document, but many of the phrases and characters in the document may have come from others. Traditionally, this issue has been solved using embedded references throughout the document but, even then, the ownership of such references is disputable; does the referenced text belongs to the original author or does it belong to the author making the reference? If an originator writes down information, it is true that only the originator can modify that information? What if one originator

[2]We note that in referring to homomorphic encryption, we assume no information leakage, but as with most cryptographic systems, this is not typically completely correct. Therefore, this would have to be addressed.

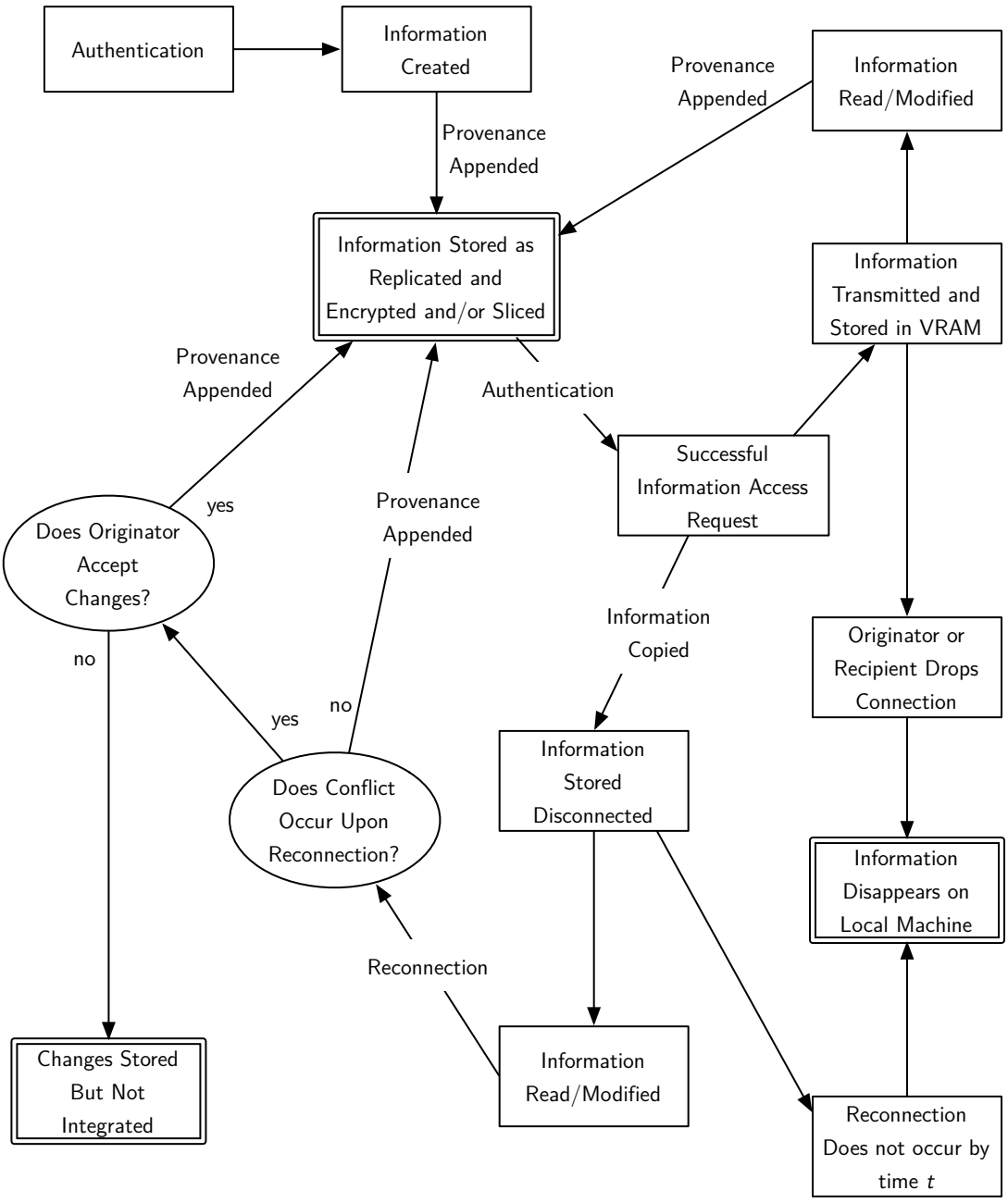

Figure 1: State flow diagram of the model.

changes the information in that document? How does the information provided by the other change, since context has now changed? The issue comes up in many cases, in terms of provenance, version control, and redaction, and as such, the complexity of the issue must be captured somehow.

This challenge is exacerbated by the fact that new information may emerge that modifies the conclusions recorded in the original reference. Typically, the original author has no way to redact these conclusions and, therefore, erroneous information is promulgated. This need for the ability to redact and/or modify information as necessary is especially critical in highly dynamic situations (e.g., national security) where information shared even a few hours ago may be completely outdated by recent events.

Clearly, the "published and archived file" paradigm of information sharing is inadequate to meet the needs of modern information environments.

Define "author" as "one who makes, or has made, changes to a document." We define two ownership cases for information: *atomic*

and *mosaic*. In atomic ownership, any change to a document must be approved by all authors because the fundamental meaning of the document has been changed and so any other sections not changed need to take the changes into account or reject them. In mosaic ownership, changes need not be approved by all authors.

Consider an example where mosaic ownership could be perilous. A prince asked the Delphi oracle whether he should join a military campaign. The oracle replied:

Ibis redibis nunquam per bella peribis

After receiving the response, the prince interpreted the phrase with commas:

Ibis, redibis, nunquam per bella peribis.

which translates as "you will go, you will return, never in war will you perish" However, if the prince moved the comma before "nunquam" to after that word, it has exactly the opposite meaning:

Ibis, redibis nunquam, per bella peribis.

translates as "you will go, you will never return, in war you will perish". So the meaning depends on where that particular comma is placed [1]. Were atomic ownership used, both the oracle and the prince would have had to agree to the alteration. But in mosaic ownership, the prince could put the comma wherever he wished.

As another example:

> "Two different redactors, working with the exact same guides, can come up with very different interpretations. . . . if two redactors identify the same fact as being classified, how much of the surrounding context do they also snip out with it? Even a stray preposition can give away information . . . [An example of] differently redacted documents came to me through two Freedom of Information Act requests to the same agency at about the same time. . . . two different people . . . looked at this document . . . In one, the top excerpt is deemed declassified and the bottom classified. In the other, the reverse. Put them together, and you have it all." [23]

These two examples show that any solution to the problem of determining the ownership of the document needs to support information ownership at both the atomic and mosaic levels. Moreover, depending on the type of information, mosaic ownership may need to be supported at the the levels of keystrokes, punctuation, letters, words, sentences, paragraphs, source code in a program, cells in a spreadsheet, pixels in a graphic image, and more, across a hierarchy of information representation and the systems that represent that information [16].

The underlying problem that this captures is provenance and, in particular, version control. In the redaction example above, good provenance and version control would have provided enough information for the redactor in one agency going back to the original author(s), and their intentions and context would inform all documents derived from the original. Absent that, multiple conflicting copies are inevitable.

In our framework, each individual piece of information created by an individual is owned in perpetuity by that individual. The pieces of information can be aggregated into larger collections of information but even in this aggregated form, the authors and modifiers of each piece of information can be identified. Information and information collections can be shared through the use of pointers. These pointers refer back to the original information but, under mosaic ownership, the recipients cannot alter the original information — they can only include it. Therefore, in a mosaic ownership situation, the originator continues to have the ability to redact, augment, or modify the information at will.

Each time an author shares a piece of information, the recipient's provenance information is added to the information's provenance list. In this way, a "linked list" for each shared piece of information is developed that enables the recipient to read an aggregated document and determine who authored each piece of information. A document or file prepared in this way does not consist of actual text characters. The document is a collection of (possibly nested) links, each pointing to information stored in a multitude of authors' repositories. Viewing a document consists of following these links back to the repositories and retrieving the individual pieces of data.

As we have discussed, every piece of information that is to be saved in this manner is put into non-volatile, write once, read many, enduring storage. This author repository can be considered a lifetime log of all of the information additions, deletions, changes, and shares made by the authors. Each of these piece of information is individually provenanced and addressed to enable recipients to reference it. To make changes, the author redirects links from earlier versions of the information to the newly developed information in the *author repository*. To redact information, the author points the links for this information to null.

5 DISCUSSION OF TRADEOFFS

Since a retrieved "document" is assembled for viewing from multiple author repositories for the reader, no enduring local storage of the actual document by the recipient is necessary, and indeed in most cases not storing it locally is ideal. Only a link to the assembled document would be needed. In fact, enduring local storage of an assembled document is antithetical to our paradigm because a locally stored copy of the document would not allow the authors to redact, augment, or modify their contributions.

Authors are responsible for identifying the security level of their information updates. Conceivably, the security level of a collection of information changes as new pieces of information are aggregated into the document. For a "textual" document, as keystrokes are aggregated into phrases, phrases into sentences, and sentences into documents, security will need to be considered at each step. For a document composed of imagery, audio, or video, the appropriate unit of "atomic" data granularity (e.g., a frame of video?) must be considered separately. In all cases, this "mosaic effect" will be managed by the author during the creation of the original document and by the recipients as information is aggregated from multiple authors.

Performance is something not addressed in this document, but largely assumed to be acceptable within the goals of the model, in part because our vision for this approach is envisioned to have overcome current limitations. That said, there are unquestionably performance limitations, such as those imposed by the laws of

physics, such as the speed of light limiting access times on distributed networks. These limitations on latency, and potentially also throughput, are inherent, and cannot be circumvented based on our current understanding of physics.

6 EXAMPLE APPLICATION OF THE MODEL

Suppose Alice wishes to create a Very Important Piece of Information, V, and save that information in a computer system. The requirements for V are that it needs to support very high degrees of confidentiality, integrity, and availability of the information equally. Alice, whom we call an *originator*, creates V by sitting at her highly secure terminal [16] and authenticating using techniques that provide an appropriate measure of confidence that Alice is who she claims to be [17]. Alice then begins typing V.

Beginning at the the very first keystroke, and continuing with each subsequent keystroke, the following happens:

(1) V, along with the authentication information identifying Alice, the action that Alice has taken with regard to the information, and the timestamp of the action are all digitally signed and encrypted. The latter forms the first entry in the provenance list;
(2) The ciphertext $C(V)$ is split into 4 slices, $S_i(C(V)), i = 1, \ldots, 4$; and
(3) Four replicas of each slice $S_{i,1}(C(V)), \ldots, S_{1,4}(C(V))$ are sent to 10 machines each, resulting in a total of 40 locations for the 10 replicas of each of the 4 slices of $C(V)$.

With each subsequent action, the information is also appended to each replica. Each of these replicas both stores the information in encrypted form and has physical security to guard the machines. Alice's machine itself stores none of this information — on her machine, it is present only in VRAM, for example a dumb terminal. Alice's system must support encryption, splitting, provenance, and authentication locally to enable end-to-end verifiability of the original information.

Suppose Alice wishes to share V with a number of colleagues. Alice decides the rules by which these people must authenticate themselves to obtain access to V. So, if Alice's colleagues are Bob and Carol and Dave, she shares this information by enabling access for them. They all then connect to the information by pointing their "dumb" terminals to one replica of each of the four slices.

Bob and Carol can view this information only while they are connected. But Dave is an astronaut on a mission to Mars with Erin and Frank, and communications with Mission Control are often interrupted by solar activity. He needs access to the information throughout his trip, even during those disruptions. In Dave's case, therefore, the information must be stored locally on Dave's spacecraft. On that spacecraft, Dave can authenticate himself only with the multi-party concurrence of Erin and Frank. All recipients have significant performance requirements for their access, particularly Dave who needs to receive telemetry information from Earth about potentially harmful space debris.

If Bob, Carol, and Dave simply read the document, then they are recipients. However, if Alice agrees, Bob, Carol, and Dave can also be authors by contributing to the document. As with Alice's actions, any action they take with regard to V is appended to the provenance of V. But Dave's actions are stored until he re-connects to the system. To prevent race conditions, in addition to the usual techniques for simultaneous access, Dave's situation requires that any conflicts arising from his changes be flagged for the corresponding originator to accept or reject.

Suppose that Carol's role is to perform computational analysis on V in combination with her own Very Important data that she created herself, $V_C = V_1^C V_2^C V_3^C$.[3] As with V, V_C is also encrypted, sliced, and replicated. It is combined with V to produce $V' = VV_C$. But Carol later wishes to redact V_2^C. She locates the pointer that points to V_2^C and changes it to point to a null area. Now, anyone seeing V' will see $V' = VV_1^C V_3^C$, and the data and replicas need not be updated. □

A state flow diagram of this example, demonstrating the application of these rules, is shown in Figure 2.

7 CONCLUSIONS

We have presented principles of information sharing and control that support traditional ORCON policies in a *need to share* or *responsibility to share* world. Our principles increase confidence in confidentiality through cryptography, data slicing, homomorphic encryption, and/or operations only in "connected" modes of operation. At the same time, we embed these principles in the context of fault tolerance mechanisms that equally provide data integrity and availability to support near-real-time requirements.

It is enlightening to consider how this model would function without the design decisions that we prescribe. For example, suppose that the timestamp of the original action were not digitally signed — in that case, neither origin nor authenticity could not be established at all. Alternatively, if the model did not split the data, should the confidentiality of even a single replica be compromised, the entire dataset would be compromised. And if the model *did not replicate* the slices, should a single system be lost the availability and integrity (but not necessarily the confidentiality) to access and/or recover the data would be lost. Of course, the union of the last two consideration suggests that if both splitting and replication were not used, the system could fail in any of the three primary ways — confidentiality, integrity, or availability – with the compromise or loss of a single component system.

Another interesting approach is to examine the information flow properties of this model in information theoretic terms. This would allow us to provide alternate definitions of the properties, and perhaps others of interest to specific environments. From these, we could reason about constraints on access and, more generally, information flow.

While we address practical aspects of the model, there are also a number of elements that we do not cover in this paper, including implementation details such as the tools that could be used to build the system, the user interface, and what access controls might look like.

There are aspects to both ORCON and also our model that are notionally very similar to what is commonly described as *digital rights management (DRM)*. DRM has probably historically been seen by the general public most frequently as a technique used by the owners or distributors of copyrighted, creative works, such as music and movies, for identifying and preventing theft of that

[3] Note that secure multi-party computation might be used for this.

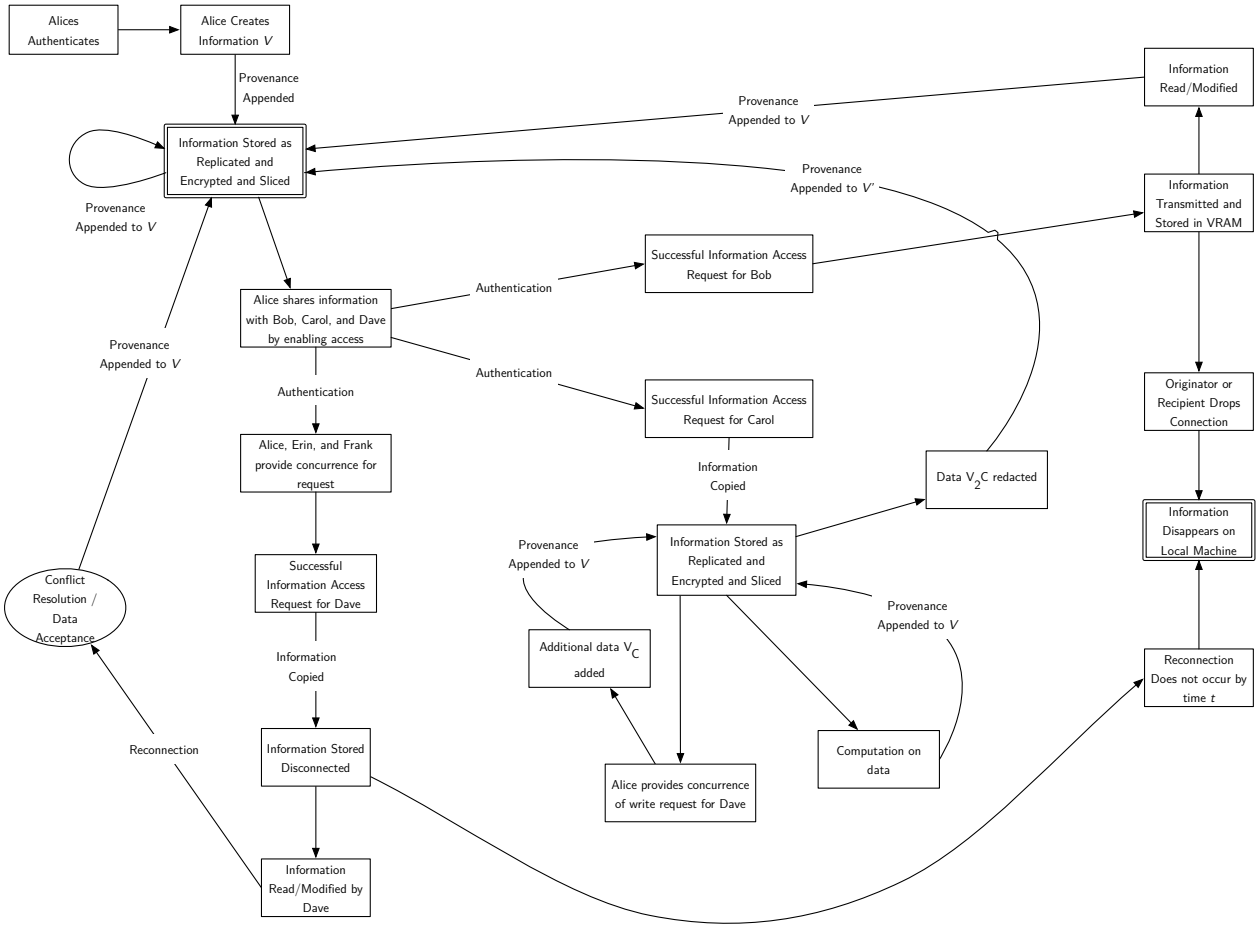

Figure 2: State flow diagram of the example.

work. DRM also has significant use in tracking document theft, and indeed, numerous commercial products exist that have this functionality.

There are also aspects of our model that bear similarity to hypertext, distributed version control systems, or distributed filesystems that enable "disconnected access." In addition, we note, given the attention that blockchains have received even in the popular press these days, one might envision certain aspects of our model, namely the integrity ledger, could be implemented by "permissioned" or "private" blockchains (essentially, a distributed, Byzantine fault tolerant [9], Merkle tree [11] of cryptographic hashes with access controls) – that is, blockchains that can be written to by via access control permissions, rather than requiring "proof of work" like the blockchain used in the popular bitcoin "cryptocurrency" [12]. All of these tools may form pieces of a possible implementation, although we do not examine details of what an implementation using these tools might look like. Given that we do not address implementation details, we also do not address the relationship between this model and side channels or covert channels – a potential avenue for future work.

Also beyond the scope of this paper is a discussion of ethics, for the same reasons as mentioned earlier. It is true that the focus of our model is one that seeks to enable a kind of "extreme traceability," which is antithetical to individual privacy. This again, is by design due to the intended application of our model to extremely security-sensitive systems. A system that provided extreme privacy and extreme traceability is an interesting notion to consider that cuts to potential current and future definitions of "identity" and how identity will be established – another potential avenue for future work. On the other hand, we do discuss practical aspects of our own model, beyond that of accidental access or misplaced trust, as it is very much a design goal of our model to see this approach be put into practice, with the full range of practical tradeoffs carefully considered.

ACKNOWLEDGEMENTS

This work was supported in part by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. It is also supported in part by the National Science Foundation under Grant Numbers CCF-1018871, ACI-1246061, and DGE-1303211.

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of any of the employers or sponsors of this work.

REFERENCES

[1] Nathaniel Bailey. 1721. *An Universal Etymological English Dictionary.* London : Printed for T. Osborne [and 27 others].

[2] David Elliott Bell and Leonard J. LaPadula. 1975. *Secure Computer System: Unified Exposition and Multics Interpretation.* Technical Report EST-TR-75-306. Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, MA.

[3] Ken Biba. April 1977. *Integrity Considerations for Secure Computer Systems.* Technical Report MTR-3153. MITRE Corporation, Bedford, MA.

[4] Matt Bishop. 2003. *Computer Security: Art and Science.* Addison-Wesley Professional, Boston, MA.

[5] Roxana Geambasu, Tadayoshi Kohno, Amit Levy, and Henry M Levy. 2009. Vanish: Increasing Data Privacy with Self-Destructing Data. In *Proc. of the 18th USENIX Security Symposium.*

[6] Craig Gentry. 2009. *A Fully Homomorphic Encryption Scheme.* Ph.D. Dissertation. Stanford University.

[7] James Gosler. 2005. The Digital Dimension. In *Transforming U.S. Intelligence,* Jennifer E. Sims and Burton L. Gerber (Eds.). Georgetown University Press, 96–114.

[8] Richard Graubart. 1989. On the Need for a Third Form of Access Control. In *Proceedings of the 12th National Computer Security Conference.* 296–304.

[9] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4, 3 (July 1982), 382–401.

[10] Steven B. Lipner. 1982. Non-Discretionary Controls for Commercial Applications. In *Proceedings of the 1982 IEEE Symposium on Security and Privacy.* Oakland, CA, 2–10.

[11] Ralph C. Merkle. 1980. Protocols for Public Key Cryptosystems. In *IEEE Symposium on Security and Privacy.* IEEE, 122–122.

[12] Satoshi Nakamoto. 2009. Bitcoin: A Peer-to-Peer Electronic Cash System. http://www.bitcoin.org/bitcoin.pdf. (May 24, 2009).

[13] Office of the Director of National Intelligence. 2008. United States Intelligence Community Information Sharing Strategy. http://www.dni.gov/reports/IC_Information_Sharing_Strategy.pdf. (Feb. 22 2008).

[14] Jaehong Park and Ravi Sandhu. 2002. Originator Control in Usage Control. In *Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks.* IEEE, 60–66.

[15] Jaehong Park and Ravi Sandhu. 2002. Towards Usage Control Models: Beyond Traditional Access Control. In *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies (SACMAT).* 57–64. https://doi.org/10.1145/507711.507722

[16] Sean Peisert, Ed Talbot, and Matt Bishop. 2012. Turtles All The Way Down: A Clean-Slate, Ground-Up, First-Principles Approach to Secure Systems. In *Proceedings of the 2012 New Security Paradigms Workshop (NSPW).* Bertinoro, Italy, 15–26.

[17] Sean Peisert, Ed Talbot, and Tom Kroeger. 2013. Principles of Authentication. In *Proceedings of the 2013 New Security Paradigms Workshop (NSPW).* Banff, Canada, 47–56.

[18] Raluca Ada Popa, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan. 2012. CryptDB: Processing Queries on an Encrypted Database. *Commun. ACM* 55, 9 (2012), 103–111.

[19] Raluca Ada Popa, Emily Stark, Jonas Helfer, Steven Valdez, Nickolai Zeldovich, M Frans Kaashoek, and Hari Balakrishnan. 2014. Building Web Applications on Top of Encrypted Data Using Mylar. In *Proceedings of the 11th Symposium on Networked Systems Design and Implementation (NSDI).* 157–172.

[20] Adi Shamir. 1979. How to Share a Secret. *Communications of the ACM (CACM)* 22, 11 (1979), 612–613.

[21] The White House. 2012. National Strategy for Information Sharing and Safeguarding. http://www.whitehouse.gov/sites/default/files/docs/2012sharingstrategy_1.pdf. (Dec. 2012).

[22] Tom Walcott and Matt Bishop. 2004. Traducement: A Model for Record Security. *ACM Transactions on Information and System Security (TISSEC)* 7, 4 (Nov 2004), 576–590.

[23] Alex Wellerstein. 2013. The Problem of Redaction. http://nuclearsecrecy.com/blog/2013/04/12/the-problem-of-redaction/. (April 12, 2013).

[24] Andrew Chi-Chih Yao. 1986. How to Generate and Exchange Secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science.* IEEE, 162–167.

APPENDIX

THEOREM 1. *If a system initially satisfies all four preconditions, then the system satisfies all four preconditions after any sequence of applications of the five rules.*

PROOF. Induction Basis: Let s_0 be the initial state of the system. By the theorem statement, the system satisfies all the preconditions.

Induction Hypothesis: In states $s_0, \ldots, s_i$, the system satisfies all four preconditions.

Induction Step: Assume a system meets preconditions 1, 2, 3, and 4 in the state s_i. We consider each rule separately.

If the creation rule is applied, new data is created. The provenance list contains user u (satisfying precondition 2). The provenance list contains no viewers as no-one has viewed the data (satisfying precondition 3). The data, including the provenance list, is replicated $3f + 1$ times (satisfying precondition 1). The data is protected using encryption, physical security means, or data slicing (satisfying precondition 4). Thus, the new state s_{i+1} satisfies all four propositions, as claimed.

If the alteration rule is applied, the data is altered. The identifier of the user is added to a record in the provenance list (satisfying precondition 2), the viewer list is unaltered (satisfying precondition 3), and the modified data is replicated $3f + 1$ times (satisfying precondition 1) and is protected using encryption, physical security means, or data slicing (satisfying precondition 4). Thus, the new state s_{i+1} satisfies all four propositions, as claimed.

If the computation rule is applied, the data is not altered. Hence the provenance list contains all the authors of the data, and precondition 2 is satisfied. However, the data has been "viewed" by another person (the one invoking the computation) and so a record including the identification of the viewer is added to the provenance list, satisfying precondition 3. The replication of the data across $3f + 1$ machines has not changed, so precondition 1 is still satisfied, and data is encrypted during computation and erased after computation, satisfying precondition 4. Thus, the new state s_{i+1} satisfies all four propositions, as claimed.

If the connected access rule is applied, data is viewed but not changed. As before, the provenance list contains all the authors of the data, and precondition 2 is satisfied. However, the data has been "viewed" by another person (the one invoking the computation) and so a record including the identification of the viewer is added to the provenance list, satisfying precondition 3. The replication of the data across $3f + 1$ machines has not changed, so precondition 1 is still satisfied, and data not stored locally, trivially satisfying precondition 4. Thus, the new state s_{i+1} satisfies all four propositions, as claimed.

If the disconnected access rule is applied, data is accessed but not changed. The data has been "viewed" by another person (the one invoking the computation) and so a record including the identification of the viewer is added to the provenance list, satisfying precondition 3. The replication of the data across $3f + 1$ machines has not changed, so precondition 1 is still satisfied, Data is stored locally, and is encrypted and "time bombed," thus satisfying precondition 4. Initially, the data is not altered, so the provenance list contains the same set of authors as it did before the application of the rule. If reconnection occurs, the provenance list is updated

accordingly, satisfying precondition 2. Thus, the new state s_{i+1} satisfies all four propositions, as claimed.

Thus, if all preconditions hold in state s_i, then they hold in state s_{i+1}, completing the induction and proving the theorem.

□