

APPROVAL SHEET

Title of Thesis: POLICIES FOR OBLIVIOUS CLOUD STORAGE USING
SEMANTIC WEB TECHNOLOGIES

Name of Candidate: Vaishali Narkhede
MS Computer Science, MAY-
2016

Thesis and Abstract Approved: _____
Dr. Karuna P Joshi
Research Assistant Professor
Department of Computer Science and
Electrical Engineering

Date Approved: _____

ABSTRACT

Title of Thesis: Policies For Oblivious Cloud Storage Using Semantic Web Technologies

Vaishali Narkhede, MS Computer Science, 2016

Thesis directed by: Dr. Karuna P Joshi,
Research Assistant Professor
Department of Computer Science and
Electrical Engineering

Consumers want to ensure that their enterprise data is stored securely and obliviously on the cloud, such that the data objects or their access patterns are not revealed to anyone, including the cloud provider, in the public cloud environment. We have created a detailed ontology describing the oblivious cloud storage models and role based access controls that should be in place to manage this risk. Our colleagues at USNA have developed an algorithm to store cloud data using oblivious data structure.

We have also implemented the ObliviCloudManager application that allows users to manage their cloud data by validating it before storing it in an oblivious data structure. This application uses role based access control model and collection based document management to store and retrieve data efficiently. Cloud consumers can use our system to define policies for storing data obliviously and manage storage on untrusted cloud platforms, even if they are not familiar with the underlying technology and concepts of the oblivious data structure.

**POLICIES FOR OBLIVIOUS CLOUD STORAGE
USING SEMANTIC WEB TECHNOLOGIES**

by
Vaishali Narkhede

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
2016

*Dedicated To my father, Ravindra Narkhede, who has put tremendous efforts to make me
what I am today
and
To my mother, Shalini Narkhede, for all the love and being a constant support to me and
my family*

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the University of Maryland, Baltimore County for letting me fulfill my dream of being a student here. I would also like to thank the Computer Science Department for giving me an opportunity to write an honors thesis. To my committee, Dr. Tim Finin, Dr. Yelena Yesha and Dr. Seung Geol Choi (USNA), I am extremely grateful for your assistance and suggestions in my project. To the staff at the Computer Science Department. To my fiance Bhushan Pawar, for helping me survive all the stress and not letting me give up, and Dr. Adam Aviv (USNA), Dr. Daniel Roche (USNA), Ankur Padia for helping me understand different concepts in my project. Most of all, I am fully indebted to Dr. Karuna P Joshi, my advisor and mentor, for her understanding, patience, enthusiasm and encouragement and for being such a fantastic advisor.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	vi
Chapter 1 INTRODUCTION	1
1.1 Motivation	1
1.2 Thesis Statement	2
1.3 Thesis Overview	3
Chapter 2 RELATED WORK	5
2.1 Semantic Ib Technologies	5
2.2 Oblivious RAM / oblivious data structures	7
Chapter 3 OBLIVIOUS CLOUD STORAGE ONTOLOGY	9
3.1 Usergroup, User and Access Permissions	10
3.2 Collection and Document	11
3.3 EncryptionPolicy and KeyManagement	12
3.4 StoragePolicy and related classes	15

3.5	AuditPolicy and LoggingPolicy	17
3.6	WhitelistedCloudProvider and WhitelistedVendors	17
3.7	SWRL Rules	17
Chapter 4	OBLIVIOUS DATA STRUCTURE	21
4.1	Brief Overview	21
4.2	Security Guarantees	22
4.2.1	Bounded History independence:	22
4.2.2	Secure deletion:	23
4.3	System Components	23
4.3.1	vORAM	23
4.3.2	HIRB	24
Chapter 5	OBLIVIOUS CLOUD STORAGE MANAGER	26
5.1	Architecture	26
5.2	Design	28
5.3	Implementation	29
Chapter 6	CONCLUSION	37
6.1	Future Work	37
6.2	Conclusion	38
Appendix A	APPENDIX	40
	REFERENCES	135

LIST OF FIGURES

3.1	Ontology describing relation between Oblivious Cloud Storage Classes. . .	10
3.2	Oblivious Cloud Ontology classes - Usergroup and User.	11
3.3	Oblivious Cloud Ontology classes - Collection and Document.	12
3.4	Oblivious Cloud Ontology classes - EncryptionPolicy and KeyManagement.	13
3.5	Oblivious Cloud Ontology classes - StoragePolicy, ModificationPolicy, ChunkingPolicy and DeletionPolicy.	15
3.6	Oblivious Cloud Ontology classes - AuditPolicy, LoggingPolicy, Ac- cessLevel WhitelistedCloudProvider and WhitelistedVendors.	16
4.1	A single vORAM bucket with partial blocks.	24
4.2	A sample vORAM state with partial blocks with id0, id1, id2, id3: Note that the partial blocks for id0 are opportunistically filled up the vORAM from leaf to root and then remaining partial blocks are placed in the stash. .	25
5.1	Oblivious Cloud Manager Architecture.	27
5.2	Oblivious Cloud Manager Login Dashboard.	28
5.3	Oblivious Cloud Manager Modules.	29
5.4	Oblivious Cloud Manager Document Dashboard.	30
5.5	Oblivious Cloud Manager Add/Edit Document.	31

5.6	Oblivious Cloud Manager Document Upload message with vORAM Key (for demo purpose only).	32
5.7	Oblivious Cloud Manager Storage Policy Dashboard.	33
5.8	Oblivious Cloud Manager Add/Edit Collection.	33
5.9	Oblivious Cloud Manager Add/Change Encryption Policy.	34
5.10	Oblivious Cloud Manager Add/Change Storage Policy.	35
5.11	Oblivious Cloud Manager Add/Change Chunking Policy and Deletion Policy.	35
5.12	Oblivious Cloud Manager User Dashboard.	36
5.13	Oblivious Cloud Manager UserGroup assign permissions	36

Chapter 1

INTRODUCTION

An efficient cloud storage service provides, at a minimum, high availability and reliability along with assured data integrity, retrieval and confidentiality (37). While such services are attractive for their cost savings and rapid provisioning/scaling features; privacy and security of cloud data remains a concern for most consumers (44). and a key barrier in adoption of the cloud.

1.1 Motivation

Cloud consumers, like federal agencies and business organizations, need a secure environment for their corporate data. Public cloud infrastructure is attracting a lot of organizations to migrate to cloud for various reasons such as cost, maintenance, scalability etc. However, there are security concerns and to be addressed before the customers readily want to outsource their sensitive data. Moreover, the customers may want some advanced features so that the cloud storage can be effectively used in the organizational setting. In summary, the cloud storage should provide the following features and security guarantees (at least):

- Confidentiality: The cloud storage provider does not learn any information about customer data.

- Availability: Customer data is accessible from any machine and at all times.
- Integrity: Any unauthorized modification of customer data by the cloud storage provider can be detected by the customer while retaining the main benefits of a public storage service.
- Reliability: Customer data is reliably backed up.
- Data sharing: Customers can share their data with trusted parties.
- Efficient retrieval: Data retrieval times are comparable to a public cloud storage service.

While storing their data on cloud, consumers usually regard cloud providers under two main adversary models Honest But Curious (HBC) adversary model (44), where providers run the programs and algorithms correctly, but might look at the information passed between entities; and the malicious adversary model (44), where providers behave in an arbitrary manner. In this paper, we look at the malicious adversary model where the cloud storage provider is not completely trusted by the consumer. In this use case, cloud consumers want their cloud data as well its usage to remain private and so along with encryption of their data objects, they also want the access patterns to be oblivious to the provider. Various approaches like Oblivious RAM (ORAM) (1; 2; 3; 4; 5; 6; 7) and Oblivious Storage (40) have been proposed to address obliviousness of data.

1.2 Thesis Statement

This thesis investigates following claim:

”Storing data on cloud obliviously keeping it’s confidentiality and integrity along with high availability and reliability and using semantic web technologies to implement security policies defined for oblivious cloud storage”

Cloud based storage services are increasingly being used by consumers for their enterprise storage needs. These often contradicting requirements results in research challenges that this thesis aims to address. I have conducted a comprehensive study to review the various solutions in place to store data securely in such a way that even access patterns of data should not be revealed to anyone including cloud provider in public cloud environment and have determined the oblivious storage models and role based access controls that should be in place to manage the risk. Based on this study, we have developed an ontology describing the cloud security controls, threats and compliances. I have also developed an application that manages the storage on cloud; it validates the data before storing data in oblivious map data structure defined. It uses a role based access control model and collection based document management to store and retrieve data efficiently. With the help of ontology defined for oblivious cloud storage we intend to use it to detect if there are any contradictions in the constraints defined by policies in ontology to the data stored using the application. Cloud consumers can use this system to define oblivious storage policies and manage storage on untrusted cloud even if they are not familiar with the underlying technology and concepts of oblivious data structure implementation.

1.3 Thesis Overview

Chapter 2 does a survey of previous work relevant to the task described in this thesis. I briefly mention the contributions made in the area of oblivious cloud storage and establish the importance of the problem at hand. Chapter 3 provides details of oblivious cloud storage ontology we have developed and rules I have implemented for policy enforcement. Chapter 4 describes the oblivious map data structure. Chapter 5 begins with a description of the architecture of the prototype we have designed for implementing oblivious cloud storage. It follows with the details of how it is implemented and description of each module in it.

Finally, chapter 6 mentions the conclusions drawn from this work and comments on the future scope in this area.

Chapter 2

RELATED WORK

I have reviewed papers on various oblivious data structures available for high level privacy in cloud storage. I have also referred to the National Institute of Standards and Technology (NIST) (14; 41) and Cloud Security Alliance (CSA) (25) guidelines for secure cloud storage and ontologies that define cloud security and compliance. The main focus of my research is to be able to ensure secure deletion of cloud data, for this purpose we have considered the NIST Cloud Deletion Use Case to securely Erase Data Objects in Cloud. (17).

2.1 Semantic Ib Technologies

Ontology can be defined as an act of describing concepts (classes) in a domain with reference to the characteristic of each concept (class) elaborating its features, attributes and roles and role restrictions (22). It can be said that Ontology is an overall collection of entity instances of the classes, which compose a base for knowledge (22). An ontology contains classes that represent concepts organized in taxonomies to define superclass-subclass hierarchy, relations that represent association between concepts, attributes to describe the features of the concepts, instances that signify elements in an ontology (24).

I have used Semantic Web technologies to develop the oblivious cloud storage ontol-

ogy and prototype development, since they enable us to build the vocabulary (or ontology) using standardized languages that support our design requirements, which include interoperability, sound semantics, Web integration, and the availability of tools and system components. Semantic Web enables data to be annotated with machine understandable meta-data, allowing the automation of their retrieval and their usage in correct contexts. Semantic Web technologies include languages such as Resource Description Framework (RDF) (46) and Web Ontology Language (OWL) (18) for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions. OWL Semantic Web knowledge can also be encoded in rule format using several approaches, including N3-logic rules, SWRL (Semantic Web Rule Language) rules (47) and RIF, the new W3C standard for Rule Inter-change Formalism. These technologies can be used to provide common semantics of Service information and policies enabling all agents who understand basic Semantic Web technologies to communicate and use each others data and Services effectively.

Our most fundamental requirement is for a representation that supports inter-operability at both the syntactic and semantic levels. The OWL (18) language has a well-defined semantics that is grounded in first order logic and model theory. This allows programs to draw inferences from OWL expressions with the assurance that the subsequent interpretation is sound. An important advantage for OWL over many other knowledge-based systems languages is that there are well defined subsets (or profiles, as they are called in OWL 2) that guarantee sound and complete reasoning with various levels of complexity (e.g., N2ExpTime for OWL 2 DL). Moreover, there are also profiles that are tuned to work well with popular implementation technologies, e.g., OWL QL for databases and OWL RL for rule-based systems.

A second design requirement is for a language that is designed to integrate well with the Web, which has become the dominant technology for today's distributed information

systems. OWL is built on basic Web standards and protocols and is evolving to remain compatible with them. It is possible to embed RDF and OWL knowledge in HTML pages and several search engines (including Google) will find and process some embedded RDF. RDF is also compatible with Microdata, a Web Hypertext Application Technology Working Group HTML specification that is used to nest semantic statements within existing content on web pages. Microdata has been adopted by Schema.org, a collaboration by Google, Microsoft, and Yahoo!, and has been used to define a number of basic ontologies that are being supported by search engines.

2.2 Oblivious RAM / oblivious data structures

The goal of oblivious data structures (ODS) is to protect the access pattern from an observer such that it is impossible to determine which operation is occurring, and on which item. The seminal work on the topic is by Goldreich and Ostrovsky (1), and since then, many works have focused on improving efficiency of Oblivious RAM (ORAM) in the space, time, and communication cost complexities (For example (2; 3; 4; 5; 6; 7) are few implementations for the same; see the references section).

There has been work done on various types of solutions for storing data where cloud providers are untrusted. For instance Hybrid cloud model separates trusted components from untrusted components and host the trusted components in private cloud and storing untrusted content on public cloud infrastructure. For example, Zhang et. al. (10) and others (11) describe such a hybrid cloud scenario in their paper. In the oblivious storage (OS) model, proposed by Boneh et al. (40), the storage provided by cloud providers is viewed as a collection of key-value pairs and the query and update operations are viewed in terms of operations dealing with key-value pairs, also call items. An Oblivious Storage solution is oblivious in this context if an HBC polynomial-time adversary is unable to distinguish

between the (obfuscated) versions of two possible access sequences of equal length and maximum set size, which are polynomially related, beyond a negligible probability.

Wang et al. (8) achieved ODS for maps, priority queues, stacks, and queues much more efficiently than previous works or naive implementation of the data structures on top of ORAM. I have used ODS map system (31) developed by research partners at USNA, it improves the efficiency over Wang et al. (8) by introducing an ORAM that admits variable-size blocks and also introducing a new randomized B-tree where the node size is variable and the height of the tree is fixed to reduce the round complexity. Please refer to (31) for the detailed design and implementation of ODS map I have used to store data securely.

Chapter 3

OBLIVIOUS CLOUD STORAGE ONTOLOGY

I have designed Ontology such that each data item irrespective of its type is stored as a document and every document belongs to only one collection. So a collection has one or more documents which share similar security requirements and classify under same group. All the security policies are applied to Collection level, and all the documents in that collection automatically gets the assigned those policies. This saves time on selecting security policies every time a new document is to be added in system. Based on cloud computing security guidelines from NIST (14)(41) and Cloud Security Alliance (25), we have developed oblivious cloud storage ontology which defines classes for Document, Collection, EncryptionPolicy, StoragePolicy, AuditPolicy, LoggingPolicy, KeyManagement, DeletionPolicy, ChunkingPolicy, ModificationPolicy, Permission, User, UserGroup, WhitelistedCloudProviders, WhitelistedVendors etc.

All the classes have data properties and object properties. The data properties of class defines the relation between individuals (instances) of OWL classes to literal values whereas object properties define relation between individuals (instances) of two OWL classes. Object properties define relation between one OWL class to the other OWL class, for example hasDocuments property of Collection class defines relation between Collection class to Document class in the ontology. Figure 3.1 describes relation between different

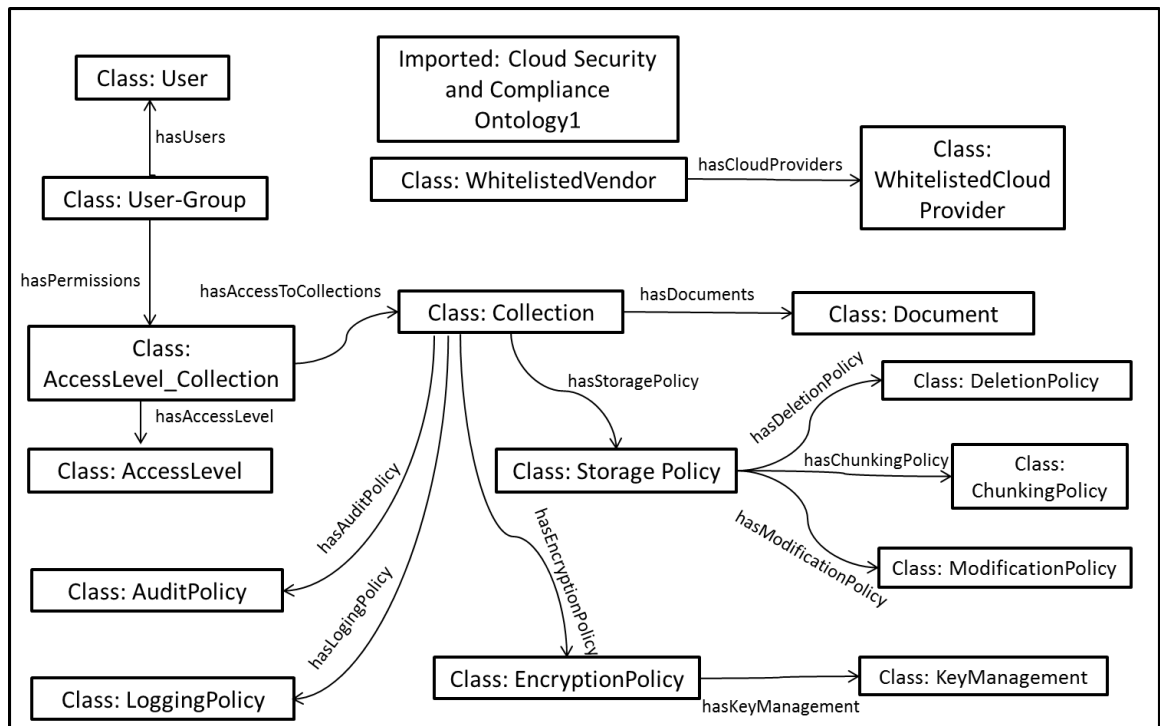


FIG. 3.1. Ontology describing relation between Oblivious Cloud Storage Classes.

Oblivious Cloud Storage Classes.

3.1 Usergroup, User and Access Permissions

Figure 3.2 shows Usergroup class, it has following data properties - groupId its a unique primary identifier, groupName a string literal for name of the user-group, It has object properties such as hasUsers contains a list of users assigned this usergroup, hasPermissions contains a list of access level permissions given for a collection for the usergroup. For example, supervisor usergroup may have WRITE access to Project2-technical collection, but just READ access to Project2-Sales collection.

AccessLevel_Collection class has hasAccessLevel property which defines level of ac-

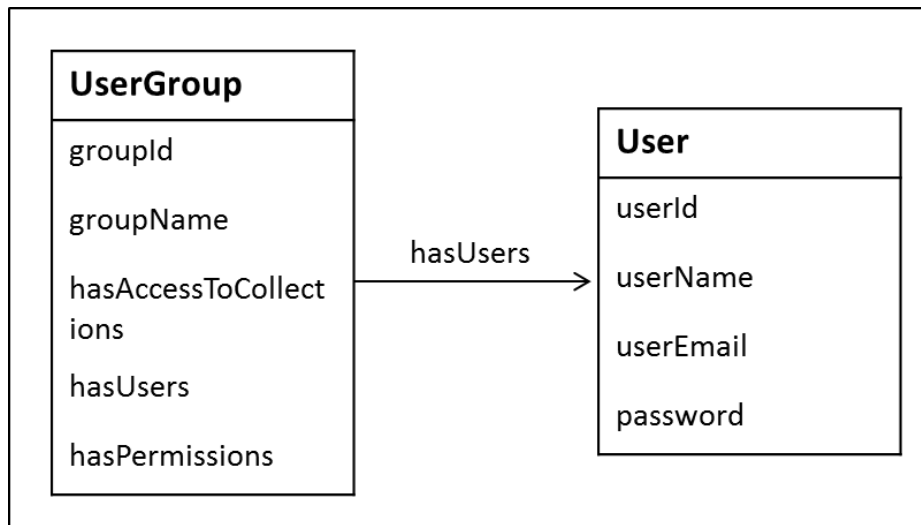


FIG. 3.2. Oblivious Cloud Ontology classes - Usergroup and User.

cess and `hasAccessToCollection` property which defines collection for which we assigning access level. Example, `proj2-read` instance of `AccessLevel_Collection` defines read access to `proj2` collection.

User class defines details specific to a user which are `userId`, a primary identifier, `userName`, email address of user and account password to access the oblivious cloud manager system.

3.2 Collection and Document

Figure 3.3 shows `Collection` class, it has following data properties - `collectionId` is a unique primary identifier, `collectionName` a string literal for name of the collection, `description` which is string literal, `rootKey` which holds key to access the collection and it's documents,

It has object properties such as `hasDocuments` contains a list of documents assigned to

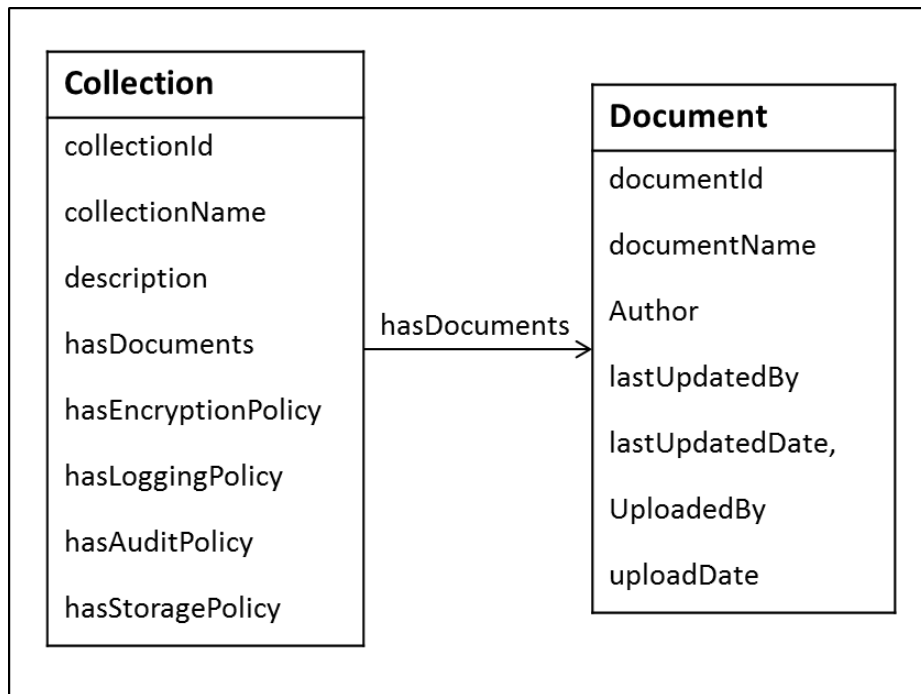


FIG. 3.3. Oblivious Cloud Ontology classes - Collection and Document.

it, `hasEncryptionPolicy` is a `EncryptionPolicy` class instance (refer section 3.3), `hasLoggingPolicy` is a `LoggingPolicy` class instance (refer section 3.5), `hasAuditPolicy` is a `AuditPolicy` class instance (refer section 3.5) and `hasStoragePolicy` is a `StoragePolicy` class instance (refer section 3.4). Document class defines details specific to a document/data file which are `documentId` its a primary identifier, document name, name of the author, last updated date, last updated by user, uploaded by user and upload date etc

3.3 EncryptionPolicy and KeyManagement

Figure 3.4 shows `EncryptionPolicy` class, it has following data properties - `encryptionAlgorithm` defines which encryption algorithm should be applied while storing the data using this `EncryptionPolicy`, for example AES, RSA, DES etc. `EncryptionType` a string literal for type of the Encryption, for example symmetric, asymmetric, basic function, hash-

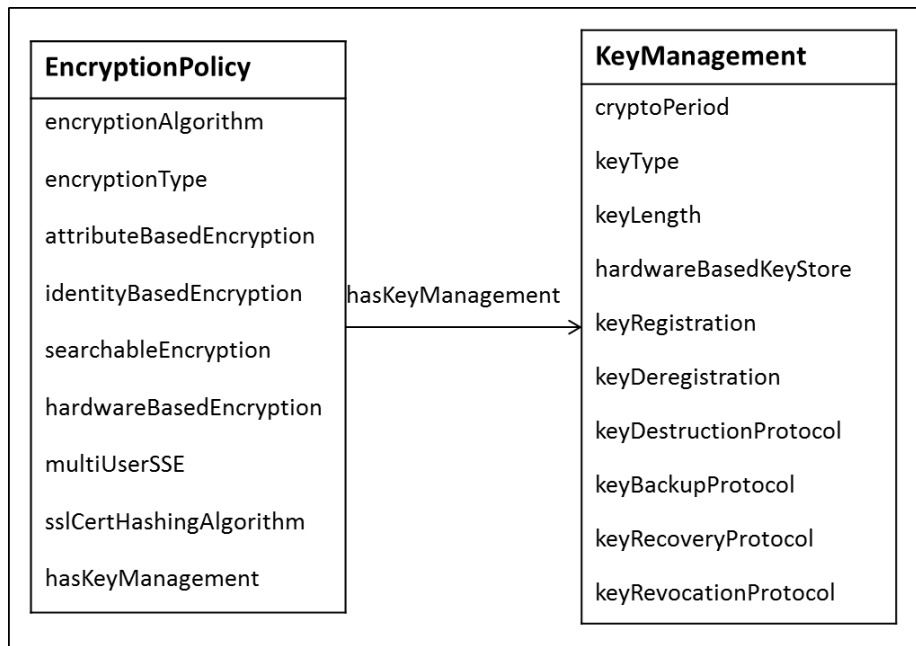


FIG. 3.4. Oblivious Cloud Ontology classes - EncryptionPolicy and KeyManagement.

ing function etc. AttributeBasedEncryption is string literal its an optional field and should be used if attribute based encryption is desired. In attribute based encryption, secret key of user and cipher-text is dependent attribute (e.g. the country he lives). IdentityBasedEncryption is string literal, its an optional field and should be used if identity based encryption is desired (e.g. encrypt and decrypt using email address of the user receiving data). SearchableEncryption a boolean literal should be selected if data should be searchable. Note that vORAM is searchable encryption HardwareBasedEncryption is string literal which is optional field and should be used if encryption is to be enforced using hardware. It has object properties such as hasKeyManagement which is a KeyManagement class instance.

For detailed attributes for key management class we have referred to NIST Key Management Recommendation guide(41). KeyManagement class has data properties as follows

-

- cryptoPeriod: This defines period after which encryption keys should be changed.

This enhances the security of the encrypted data.

- **keyType:** This defines type of the encryption key. e.g. signature key, authentication key, data encryption key, master key, transport key etc.
- **keyLength:** This defines encryption key length in bits. More the length more difficult encryption to break, but it takes long to encrypt or decrypt the data. Alternatively known as key size.
- **hardwareBasedKeyStore:** This defines if the key should be generated using hardware for password generation for example RSA Tokens.
- **keyRegistration:** This defines the process of officially recording the keying material by a registration authority.
- **keyDeregistration:** This defines protocol for the marking of all keying material records and associations to indicate that the key is no longer in use.
- **keyDestructionProtocol:** This defines protocol to remove all traces of keying material so that it cannot be recovered by either physical or electronic means.
- **keyBackupProtocol:** This defines if the keys should be backed up. Some keys should not be backed up. For example, Private Signature keys.
- **keyRecoveryProtocol:** This defines mechanisms and processes that allow authorized entities to retrieve or reconstruct keying material from key backup or archive.
- **keyRevocationProtocol:** This defines a process whereby a notice is made available to affected entities that keying material should be removed from operational use prior to the end of the established cryptoperiod of that keying material.

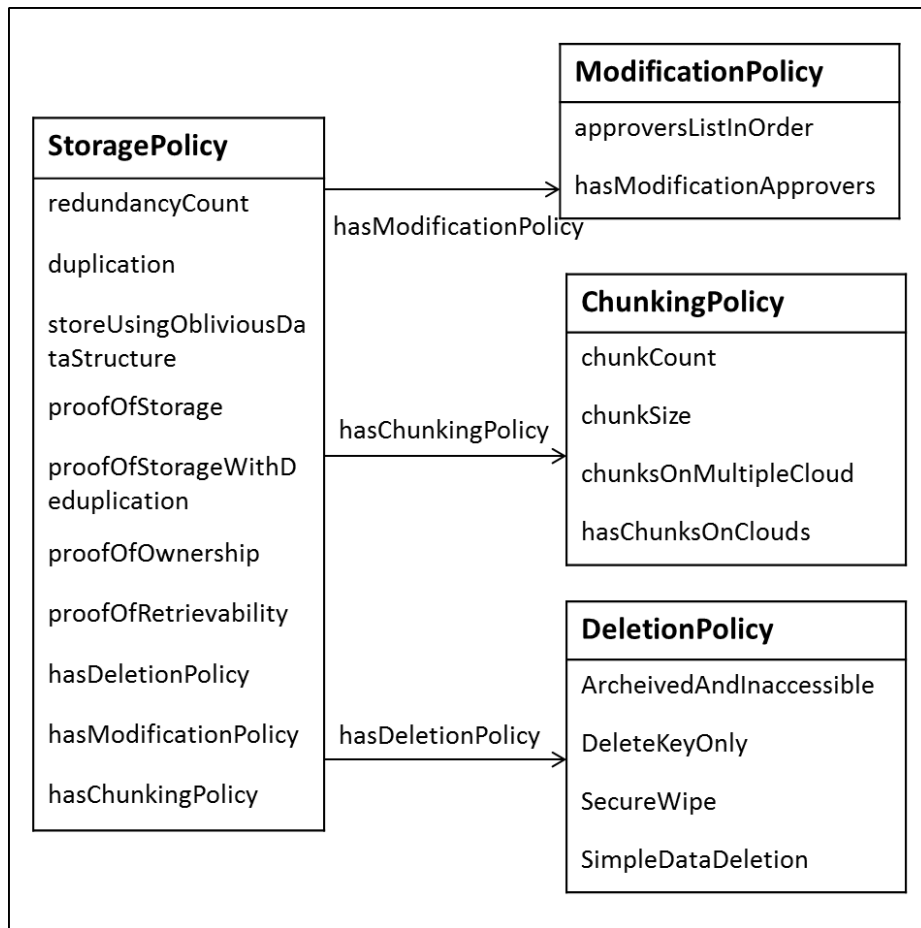


FIG. 3.5. Oblivious Cloud Ontology classes - StoragePolicy, ModificationPolicy, ChunkingPolicy and DeletionPolicy.

3.4 StoragePolicy and related classes

As shown in figure 3.5, StoragePolicy class defines details about how data should be stored using following attributes - `redundancyCount` is count of number of copies of the data to store. `Duplication` is boolean field which stores if the data should be duplicated or not. `proofOfStorage` is protocol to verify integrity of the data outsourced to the cloud. `ProofOfStorageWithDeduplication` (42) is similar but also checks if the storage is not getting duplicated. `ProofOfOwnership` is used to alleviate the cloud server from storing

multiple copies of the same data, which could substantially reduce the consumption of both network bandwidth and server storage space. ProofOfRetrievability is similar to proof of storage but offers an extra property that the client can actually recover the data outsourced to the cloud. Storage class also contains instances of ModificationPolicy, ChunkingPolicy and DeletionPolicy.

ModificationPolicy defines if the modification to data stored in cloud should go through approval workflow. Chunking Policy defines attributes for chunking the data before storing it in cloud, for e.g. chunk size, chunk count, chunk storage on different clouds etc. DeletionPolicy defines how data should be deleted, for e.g. secure wipe, archive and make it inaccessible to normal users, deleting encryption key only and simple data deletion etc.

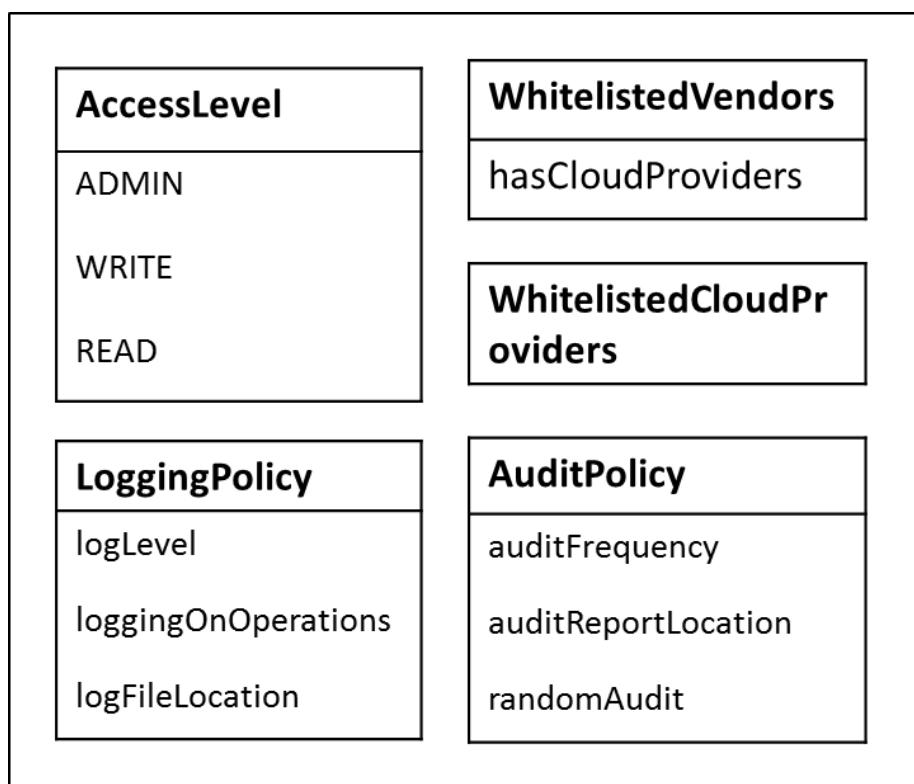


FIG. 3.6. Oblivious Cloud Ontology classes - AuditPolicy, LoggingPolicy, AccessLevel WhitelistedCloudProvider and WhitelistedVendors.

3.5 AuditPolicy and LoggingPolicy

As shown in figure 3.6, There are two more policies about auditing and logging that are applied at collection level. AuditPolicy class defines attributes such as frequency of audit, audit report file location and if random audit should be done or not. LoggingPolicy class defines log file location, log level such as INFO, ERROR, WARNING etc. Apart from log level if logging on custom operations is required etc.

3.6 WhitelistedCloudProvider and WhitelistedVendors

Now a days cloud customers tend to use one or more cloud providers for their enterprise needs and migrating existing business to cloud. Cloud vendors help them to do both tasks, they manage and provide cloud integration services to cloud customers. Whitelisted-CloudVendors class defines list of cloud vendors who are whitelisted, whereas WhitelistedCloudProvider defines list of cloud providers that are whitelisted.

3.7 SWRL Rules

The purpose of having ontology is to determine if there are any constraint violations in the Oblivious Cloud Storage Manager application. To do so I am using relational data to ontology mapping tool "RDBToOnto", and convert the database instances to RDF file, I have also wrote a simple java program which takes this RDF generated by RDBToOnto tool and adds datatype to all the instances (which RDBToOnto does not do on it's own and add rules defined in ontology) and then reason over the RDF generated to see if there are any violations to the rules defined by cloud customer. We have used relational database as backend of the application as most of the customers still comfortable with using relational databases at backend of system instead of RDF. So if we synchronize the backend data from

relational databases to map to a RDF file, so that to verify if all the constraints and policies are followed. For example, cloud customer may want to have a policy for keyLength property as per Collection, for instance documents in HR-Document-Collection to have key length of 64 whereas UserManual-Collection to have key length of 32. If the user uploads a document in HR-Document-Collection with key length other than 64, reasoned should be able to detect this violation of rule. In encryptionPolicy class, if searchable encryption is selected then encryption must also be symmetric. ORAM is by default searchable so if searchable is unchecked, it should identify as rule violation. In storagePolicy class if duplication is unchecked but redundancyCount is more than 1 then it should detect. In ChunkingPolicy either chunkCount or chunkSize should be used etc.

Below is list of some of the rules defined in ontology -

- If a chunking policy, has defined chunk size then chunk count must not be set, hence set to -1.
`ChunkingPolicy(?x), chunkSize(?x, ?y), swrlb:greaterThan(?y, 0) -> chunkCount(?x, -1)`
- If a chunking policy, has defined chunk count then chunk size must not be set, hence set to -1.
`ChunkingPolicy(?x), integer[0](?y), chunkCount(?x, ?y) -> chunkSize(?x, -1)`
- If a chunking policy, has defined chunk count to 1 then chunk on multiple cloud flag should be false.
`ChunkingPolicy(?x), chunkCount(?x, 1) -> chunksOnMultipleCloud(?x, false)`
- If we want to have custom rule, for example if chunks on multiple cloud is set to true, then hasChunkInClouds should have specific cloud provider then it can be defined as below:

ChunkingPolicy(?x), chunksOnMultipleCloud(?x, true) -¿hasChunksOnClouds(?x, Amazon)

- If a Storage policy, has defined redundancy count less than or equal to 1 then duplication flag should be set to false.

StoragePolicy(?x), redundancyCount(?x, ?z), integer[¿=1](?z)-¿ duplication(?x, false)

- If a Storage policy, has defined redundancy count greater than 1 then duplication flag should be set to true.

StoragePolicy(?x), redundancyCount(?x, ?y), integer[¿1](?y) -¿ duplication(?x, true)

- If a Encryption policy, has defined searchable encryption flag to true then encryption type should be set to symmetric.

EncryptionPolicy(?ep), searchableEncryption(?ep, ?s), equal(?s, true)-¿ encryptionType(?ep, "symmetric")

- Cryptoperiod is integer value in days.

KeyManagement(?km), keyType(?km, "Private Authentication Key") -¿ cryptoperiod(?km, 730)

- KeyManagement(?km), keyType(?km, "Public Authentication Key") -¿ cryptoperiod(?km, 365)

- KeyManagement(?km), keyType(?km, "Private Signature Key ") -¿ cryptoperiod(?km, 1095)

- KeyManagement(?km), keyType(?km, "Symmetric Data Encryption Key") -¿ cryptoperiod(?km, 730)

- KeyManagement(?km), keyType(?km, "Private Authorization Key") -¿ cryptoperiod(?km, 730)

- `KeyManagement(?km), keyType(?km, "Public Authorization Key ") -> crypto-period(?km, 730)`
- Below is example of custom rule, which defines if modification policy requires approval then list of approvers must have supervisor.

`ModificationPolicy(?mp), approvalRequired(?mp, ?ar), equal(?ar, true) -> hasModificationApprovers(?mp, SupervisorGroup1)`

Chapter 4

OBLIVIOUS DATA STRUCTURE

The oblivious cloud storage manager application uses an oblivious data structure (ODS) for map as the back-end to store files. Our colleagues at USNA have designed and implemented a new ODS map; the system details and experiment results have been reported in a separate article (31). Here is brief overview the system and it's useful security features.

4.1 Brief Overview

The main features that the oblivious map provides are as follows:

- Its a keyword-value map. In other words, it is a data structure where each item is a (keyword, value) pair, and it supports inserting an item, searching for an item containing a given keyword, deleting an item containing a given keyword. In our context, a data item corresponds to a file, with the filename as the keyword and the file contents as the value.
- It provides obliviousness. That is, the data structure will ensure that not only the raw data exchanged between the client and the server but also the client's pattern of access to the server storage reveals nothing to the server about the underlying

data or about the operations the client has performed on them (except the number of operations performed so far, which is unavoidably revealed simply by the amount of the network traffic). This a strong privacy guarantee for the client against the server. This privacy guarantee is meaningful, since its known that the access pattern, even if the actual data is encrypted, may leak sensitive information about the underlying stored data, such as keyword search queries or encryption keys (28; 29; 30).

This implementation greatly improved the efficiency of existing ODS maps. The performance of it using Amazon Web Services, and the single-operation time for a realistic database (up to 2 18 entries) is less than 1 second. This represents a 100x speedup compared to the previous system by Wang et al. (8).

4.2 Security Guarantees

In addition to obliviousness, ODS map system provides other beneficial security guarantees. In the cloud storage scenario, obliviousness will protect the clients privacy from any observer of network traffic or from the cloud server itself. *However, if the attacker compromises the client and obtains critical information such as the encryption keys used in the ODS*, all the sensitive information stored in the cloud will simply be revealed to the attacker. We call this scenario a *catastrophic attack*, and it is important to stress that this attack is quite realistic. The client machine may be stolen or hacked, or it may even be legally seized due to a subpoena. This ODS system provides the following security guarantees even against the catastrophic attack:

4.2.1 Bounded History independence:

By inspecting the internal structure of the currently existing data in the cloud after a catastrophic attack, the attacker may still be able to infer information about which items

were recently accessed or the likely prior existence of a record even if that record was previously deleted. However, if an ODS scheme provided perfect history independence, the catastrophic attacker cannot infer which sequence of operations was applied, among all the sequences that could have resulted in the current set of the data items. Interestingly, we show that it is impossible to achieve perfect history independence in our setting with a computationally limited server (31). This ODS system achieves bounded history independence; that is, the internal structure of the map system reveals only the recent operations, and old operations cannot be inferred by the structure.

4.2.2 Secure deletion:

Given that only bounded history independence is possible, the privacy of deleted data must be considered. In our system, the catastrophic attacker cannot guess information about deleted data.

4.3 System Components

Following the previous ODS system by Wang et al. (40), we achieve an ODS by constructing a tree-based data structure on a non-recursive ORAM. In order to improve efficiency and provide a better security guarantee, however, we designed a new tree-based data structure, called HIRB (history-independence randomized b-tree) and a new ORAM that admits variable size data blocks, called vORAM.

4.3.1 vORAM

The design of vORAM is based on the non-recursive version of Path ORAM (6), but it adds more flexibility by allowing each ORAM bucket to contain as many variable-size blocks (or parts of blocks) as the bucket space allows. vORAM preserves obliviousness and

maintains a small stash as long as the size of variable blocks can be bounded by a geometric probability distribution. To support secure deletion, it stores encryption keys within each bucket for its two children, and these keys are re-generated on every access, similarly to other work on secure deletion (27).

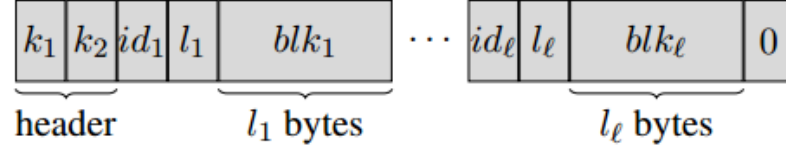


FIG. 4.1. A single vORAM bucket with partial blocks.

Each node (or bucket) in vORAM tree is split into two areas: header and data. See Figure 4.1 for a pictorial description. The header area contains two encryption keys for the two child buckets. The data area contains a sequence of (possibly partial) blocks, each preceded by a unique identifier string and the block data length. The end of the data area is filled with 0 bits. Each id_i uniquely identifies a block and also encodes the path of buckets along which the block should reside. Partial blocks share the same identifier with each length l indicating how many bytes of the block are stored in that bucket. Recovering the full block is accomplished by scanning from the stash along the path associated with id (see Figure 4.2). The main observation is that the irregularity of block sizes can be smoothed over $O(\log n)$ buckets from the vORAM root to an vORAM leaf, where n is the maximum number of blocks that vORAM contains. vORAM is the first ORAM that deals with variable size blocks, and may be of independent interest.

4.3.2 HIRB

It's a new data structure, called a HIRB tree (history independent, randomized B-tree), that satisfies all the aforementioned requirements. Conceptually, it is a fixed height B-tree

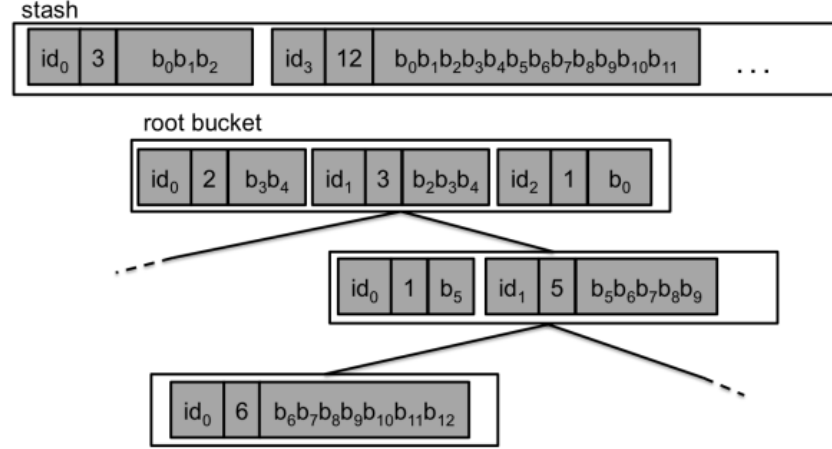


FIG. 4.2. A sample vORAM state with partial blocks with id_0 , id_1 , id_2 , id_3 : Note that the partial blocks for id_0 are opportunistically filled up the vORAM from leaf to root and then remaining partial blocks are placed in the stash.

such that when each item is inserted, the level in HIRB tree is determined by $\log n$ trials of (pseudorandom) biased coin flipping where β is the block factor. The tree may split or merge depending on the situation, but it never rotates. The fixed height of the tree, i.e. $H = 1 + \log n$, is very beneficial for efficiency. In particular, every operation visits at most $2H$ nodes, which greatly saves on padding costs, compared to previous ODS schemes.

Chapter 5

OBLIVIOUS CLOUD STORAGE MANAGER

Oblivious Cloud Manager Application is web based application, it should be used to securely store the data on cloud, it validates the data before storing data in oblivious map data structure defined. It uses a role based access control model and collection based document management to store and retrieve data efficiently. With the help of ontology defined for oblivious cloud storage, it can be used it to detect if there are any contradictions in the constraints defined by policies in ontology to the data stored using the application. Cloud consumers can use this system to define oblivious storage policies and manage storage on untrusted cloud even if they are not familiar with the underlying technology and concepts of oblivious data structure implementation.

5.1 Architecture

Figure 5.1 shows architecture diagram of the application. Its a web application which defines the policies enforcement on the documents. User accesses the application using browser and it has role based access control system for managing User module. User-group permissions are automatically given to all its users. Application back-end uses relational database structure which is mapped from the Ontology we have defined in chapter 3. Application hosts oblivious data structure (ODS) implementation described in chapter 4. So all

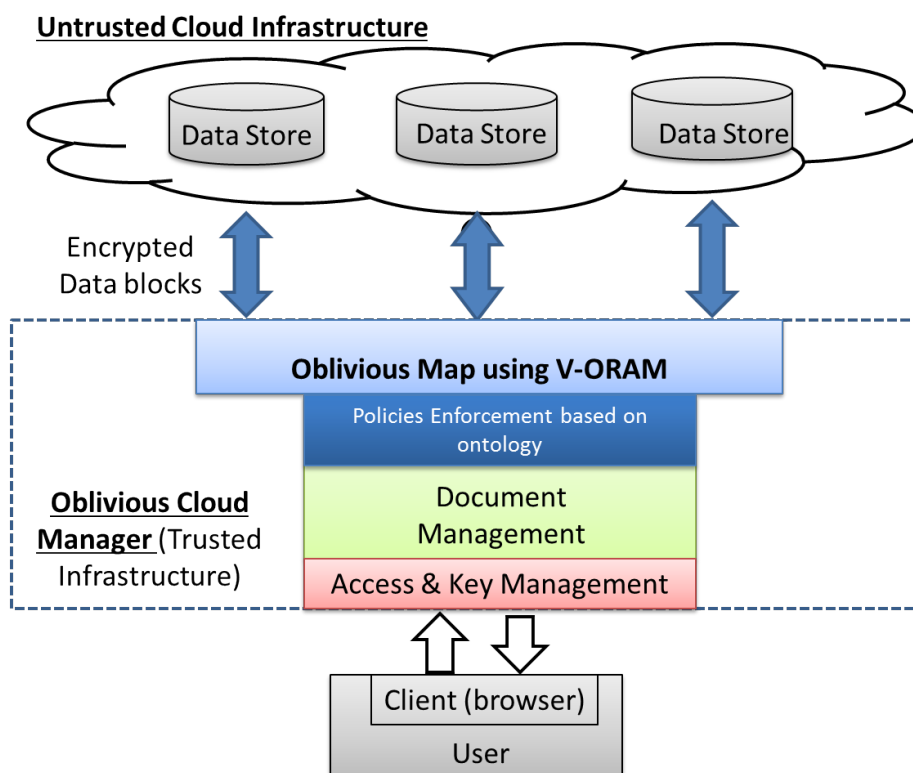


FIG. 5.1. Oblivious Cloud Manager Architecture.

the documents are stored using ODS and only encrypted blocks of the file are then sent to cloud for storage. These blocks do not require any security while communicating to cloud such as secure FTP connection or anything else as it cannot be encrypted at all without root-key of VORAM which stays with the application. The application can be deployed on in house infrastructure of cloud customer or on a trusted private cloud. Oblivious cloud storage manager uses policies defined in Ontology and stores the documents in ODS only if all the constraints in ontology are met. This application acts as a client to oblivious storage system and manages how the data is getting stored on public cloud infrastructure and it should be hosted on trusted infrastructure such as private cloud or in house infrastructure. This scenario is similar to that considered by Emil Stefanov and Elaine Shi in ObliviStore:

High Performance Oblivious Cloud Storage paper (13) and Williams et. al. in their PrivateFS system (12).

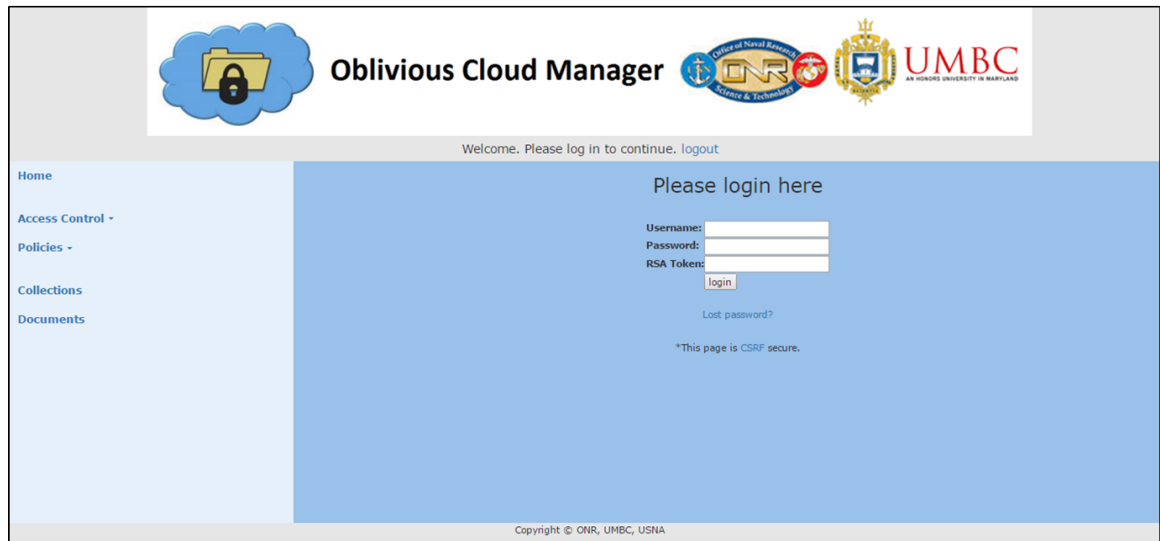


FIG. 5.2. Oblivious Cloud Manager Login Dashboard.

5.2 Design

- Every Document is stored as a part of Collection.
- Document is stored in ODS based on the policies applied to the Collection it belongs to.
- User-group will have different level of access to Collection such as read or write or admin.
- User will be assigned to a user-group, so all the permissions to that user-group are automatically given to the user.

- Access permissions can be either given to the user-group or to individual user. We can assign a single user to multiple user-groups, so that s/he can access different collections (of documents) as per her/his role.

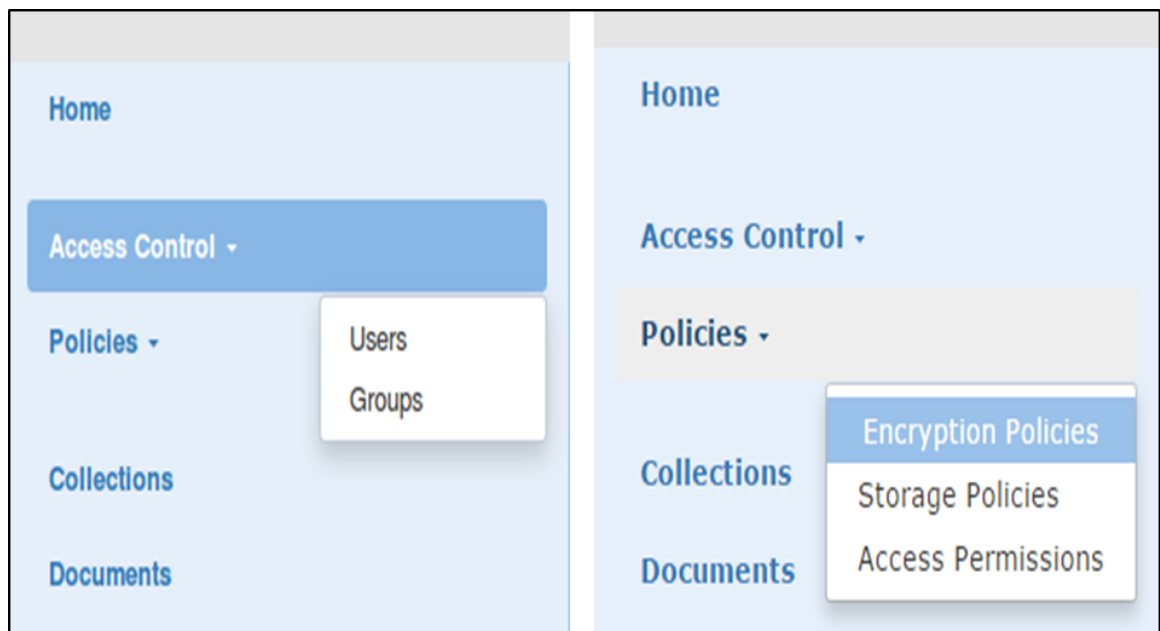


FIG. 5.3. Oblivious Cloud Manager Modules.

For example, as a programmer John should have read - write access to a projects technical specification guide, but should not necessarily have write permissions to that projects pricing document. As a manager of the project Dave, should have admin access to all project related collections, so that he can add/edit users, read write documents to all collections.

5.3 Implementation

It's a web application developed using Django Framework in Python, HTML, javascript, jQuery for UI and MySQL database in the back-end. It uses a Model-View-

Controller design pattern and comes with many in built modules such as administration module, support and easy migration to various databases.



FIG. 5.4. Oblivious Cloud Manager Document Dashboard.

Figure 5.4 illustrates the application dashboard, which has 4 different modules shown at left panel - Access Control, Policies, Collections and Document modules. Access Control has two sub modules user and group model. Policies have three sub modules for defining encryption policy, storage policy and access permission policy for collection and user-group mapping.

We can view collections as a set of documents. User-group will have access to Collections. This will be one to many relationship. User-group will have different level of access permissions on different collections. A mapping of this role based access model will be stored in access permission sub-module.

Key Management can be deployed using different types of key management mechanisms, such as 1) Storing keys in cloud itself using their key management software on segregated cloud than data cloud. 2) Storing keys at customer trusted infrastructure to

Home > Filestore > Documents > Add document

Add document

Collection: Project-ABC-Sales

Document name: ABC-Sales-2012-Report

File location: cloud VORAM

Upload date: Date: 2016-02-03 Today | Time: 20:16:55 Now |

Last modified date: Date: 2016-02-03 Today | Time: 20:16:57 Now |

Added by: admin

Last modified by: -----

Access count: 0

Docfile: Choose File download.pdf

Save and add another Save and continue editing Save

FIG. 5.5. Oblivious Cloud Manager Add/Edit Document.

have full access of keys hence data. 3) Hardware based cryptographic storage. 4) Rotating credentials for enhanced password protection.

Figure 5.9 shows add/change encryption policy form; it has all the fields defined by the ontology such as encryption algorithm, type, attribute/identity based encryption etc. It also defines which type of Key Management is supported in this encryption policy. Upon clicking on add new key Management policy a similar form with details particular to encryption key details and storage pops up. It has fields such as key length, key type, type of key storage, crypto period etc.

Figure 5.10 shows add/change storage policy form, it has all the fields defined by

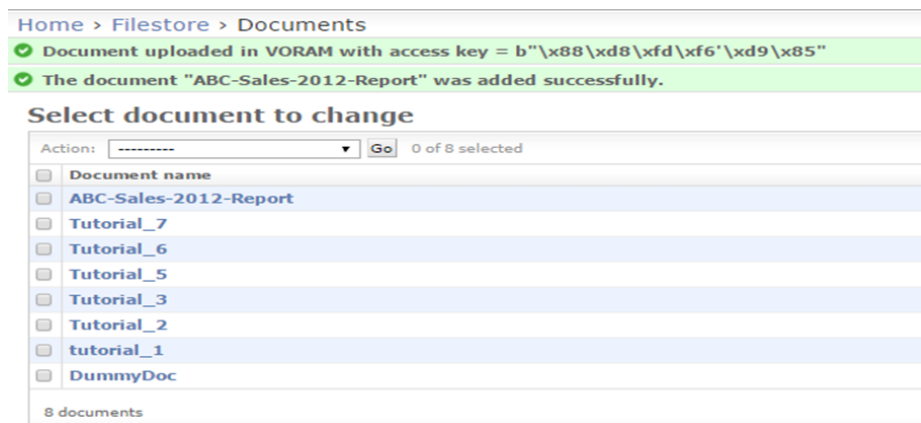


FIG. 5.6. Oblivious Cloud Manager Document Upload message with vORAM Key (for demo purpose only).

the ontology such as redundant storage, proof of storage protocol, proof of ownership etc. It also defines which type of deletion policy, modification policy, and chunking policy is supported in this storage policy. Upon clicking on add new policy form is generated with respective details.

Application is Cross-Site Request Forgery (CSRF) safe (25). CSRF is a type of attack that occurs when a malicious Web site, email, blog, instant message, or program causes a user's Web browser to perform an unwanted action on a trusted site for which the user is currently authenticated. CSRF holds potential to compromise complete web application, if victim is administrative account. CSRF attacks are target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. With this though attacker cannot get any data, he is able to execute instructions of his choice which may have serious security threats. For example, changing email address, transfer or delete of data.



FIG. 5.7. Oblivious Cloud Manager Storage Policy Dashboard.

Home > Filestore > Collections > Add collection

Add collection

Collection name:

Date published: Date: Today | Time: Now |

Description:

Rootkey:



EncryptionPolicy:

StoragePolicy:

FIG. 5.8. Oblivious Cloud Manager Add/Edit Collection.

Home > Filestore > Encryption policiys > RSA_Strict_Enc_Policy Algorithm: RSA

Change encryption policy

Encryption policy name:	<input type="text" value="RSA_Strict_Enc_Policy"/>
EncryptionAlgorithm:	<input type="text" value="RSA"/>
EncryptionType :	<input type="text" value="symmetric"/>
AttributeBased Encryption:	<input type="text" value="NO"/>
IdentityBasedEncryption:	<input type="text" value="EMAIL"/>
SearchableEncryption:	<input type="text" value="No"/>
HardwareBasedEncryption:	<input type="text" value="No"/>
MultiUserSSE:	<input type="text" value="No"/>
SslCertHashing Algorithm:	<input type="text" value="ANY"/>
Key management:	Key management policy: GeneralKeyManagement-1, Key Length: 32, Crypto Period: 120 Days  


 Delete

FIG. 5.9. Oblivious Cloud Manager Add/Change Encryption Policy.

Home > Filestore > Storage policies > Simple Storage 1

Change storage policy

Storage policy name:

Duplication:

ProofOfStorage :

ProofOfStorage WithDeduplication:

ProofOfOwnership:

ProofOfRetrievability:

DeletionPolicy:

ModificationPolicy:

ChunkingPolicy :

FIG. 5.10. Oblivious Cloud Manager Add/Change Storage Policy.

Home > Filestore > Chunking policies > Add chunking policy

Add chunking policy

Chunking policy name:

ChunkCount:

ChunkSize:

ChunksOnMultipleCloud:

ChunksOnClouds:

Hold down "Control", or "Command" on a Mac, to select more than one.

Home > Filestore > Deletion policies > Add deletion policy

Add deletion policy

Deletion policy name:

ArchivedAndInaccessible:

DeleteKeyOnly:

SecureWipe:

SimpleDataDeletion:

FIG. 5.11. Oblivious Cloud Manager Add/Change Chunking Policy and Deletion Policy.



FIG. 5.12. Oblivious Cloud Manager User Dashboard.

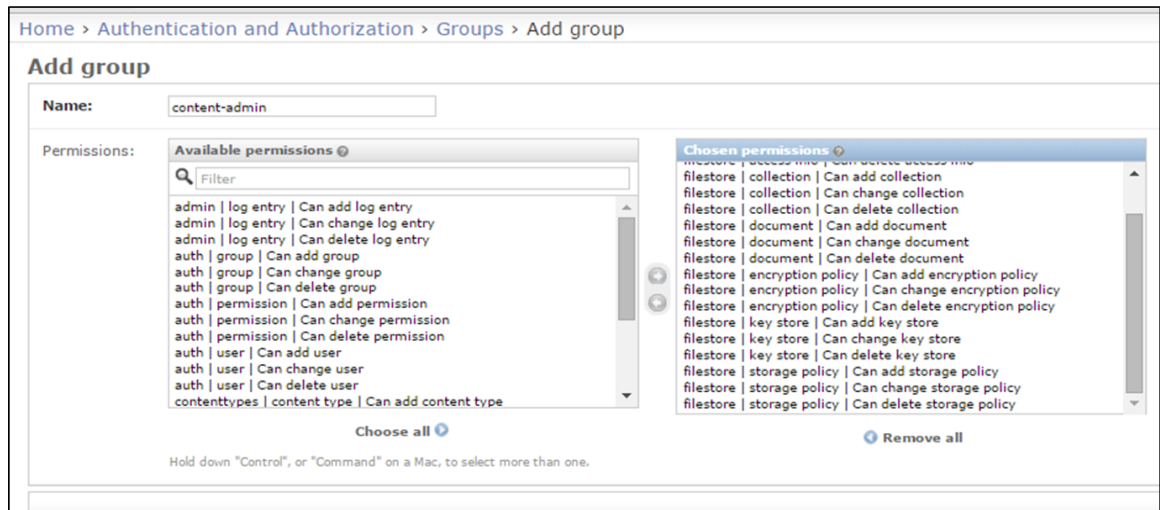


FIG. 5.13. Oblivious Cloud Manager UserGroup assign permissions .

Chapter 6

CONCLUSION

To evaluate our web application we implemented it on a trusted Amazon EC2 instance. I were successfully able to upload various types of files and store them at a defined file location obviously as encrypted blocks of data. Unauthorized users cannot access the files even if they get access to these encrypted blocks, nor can they learn anything from the access pattern structure.

I have defined rules to implement security policies using ontology, and have mapped oblivious cloud manager's data to create RDF to reason over the data store in oblivious cloud manager to check if there are any policy violations.

6.1 Future Work

As this part of my thesis I built a system which cloud customers can use to store their data obviously in cloud, Future Work for this project can be -

- **Synchronizing application with ontology**

Integrating ontology and application so that changes in ontology can reflect in application as soon as possible.

- **Multiple Clouds**

Implementing oblivious cloud storage on multiple cloud providers as ontology defines option to store data on multiple clouds for enhanced security.

- **Detailing Ontology**

Adding details to ontology to expand key management protocols, for example, keyRegistration, keyDeregistration, keyRevocation are string properties which can be expanded to new Key Management child classes.

- **Hierarchical collection**

Implementing hierarchical collection system, so that usergroups can inherit access permissions to all children collection.

- **Configurable Oblivious Data Structure**

Modifying ODS to accept various security parameters to store data in it etc. For example, encryption algorithm, key length etc.

6.2 Conclusion

I have conducted a comprehensive study to review cloud computing security guidelines for secure storage and deletion. I have reviewed many oblivious storage algorithms and their implementations. I used this study to develop a semantically rich ontology to define policies for oblivious storage. The ontology defines policies for oblivious storage in cloud that should be in place to detect if there are any constraint violations in the policies defined by the cloud consumer. I have also developed an easy to use oblivious cloud storage manager application for consumers who are planning to move their data to the cloud but are hesitant due to security concerns as they may not be aware of the security controls. I have added various constraints in policies defined in ontology and mapped the application back-end data to RDF file and reasoned over it to check if there are any constraint

violations.

Appendix A

APPENDIX

```
<?xml version="1.0"?>
```

```
<!DOCTYPE rdf:RDF [  
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >  
  <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >  
  <!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#" >  
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >  
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >  
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >  
  >  
  <!ENTITY obliviouscloud "http://userpages.umbc.edu/~  
    vaishall/ontologies/securecloudstorage.owl#" >  
>
```

```

<rdf:RDF xmlns="http://userpages.umbc.edu/~vaishal1/
  ontologies/securecloudstorage.owl#"
  xml:base="http://userpages.umbc.edu/~vaishal1/ontologies/
    securecloudstorage.owl"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:obliviouscloud="http://userpages.umbc.edu/~vaishal1
    /ontologies/securecloudstorage.owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<owl:Ontology rdf:about="http://userpages.umbc.edu/~
  vaishal1/ontologies/securecloudstorage.owl">
  <owl:imports rdf:resource="http://www.semanticweb.org/
    ontologies/2014/5/untitled-ontology-20"/>
</owl:Ontology>

```

```

<!--

```

```

////////////////////////////////////

```

```

//

```

```

// Object Properties

```

```
//
```

```
////////////////////////////////////
```

```
—>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasAccessToCollections —>
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasAccessToCollections">
  <rdfs:range rdf:resource="&obliviouscloud;Collection"/>
  <rdfs:domain rdf:resource="&obliviouscloud;Usergroup"/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasAuditPolicy —>
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasAuditPolicy">
  <rdfs:range rdf:resource="&obliviouscloud;AuditPolicy"/>
```

```

    <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
  </owl:ObjectProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasChunkingPolicy -->

```

```

<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasChunkingPolicy">
  <rdfs:range rdf:resource="&obliviouscloud;ChunkingPolicy
    "/>
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
    "/>
</owl:ObjectProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasChunksOnClouds -->

```

```

<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasChunksOnClouds">
  <rdfs:domain rdf:resource="&obliviouscloud;
    ChunkingPolicy"/>

```

```

    <rdfs:range rdf:resource="&obliviouscloud;
        WhitelistedCloudProviders"/>
</owl:ObjectProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#hasCloudProviders -->

```

```

<owl:ObjectProperty rdf:about="&obliviouscloud;
    hasCloudProviders">
    <rdfs:range rdf:resource="&obliviouscloud;
        WhitelistedCloudProviders"/>
    <rdfs:domain rdf:resource="&obliviouscloud;
        WhitelistedVendors"/>
</owl:ObjectProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#hasDeletionPolicy -->

```

```

<owl:ObjectProperty rdf:about="&obliviouscloud;
    hasDeletionPolicy">
    <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>

```

```

    <rdfs:range rdf:resource="&obliviouscloud;DeletionPolicy
    "/>
</owl:ObjectProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#hasDocuments -->

```

```

<owl:ObjectProperty rdf:about="&obliviouscloud;
    hasDocuments">
    <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
    <rdfs:range rdf:resource="&obliviouscloud;Document"/>
</owl:ObjectProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#hasEncryptionPolicy -->

```

```

<owl:ObjectProperty rdf:about="&obliviouscloud;
    hasEncryptionPolicy">
    <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
    <rdfs:range rdf:resource="&obliviouscloud;
    EncryptionPolicy"/>

```



```
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasFileLocations -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasFileLocations">
  <rdfs:domain rdf:resource="&obliviouscloud;Document"/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasKeyManagement -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasKeyManagement">
  <rdfs:domain rdf:resource="&obliviouscloud;
      EncryptionPolicy"/>
  <rdfs:range rdf:resource="&obliviouscloud;KeyManagement
      "/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasLoggingPolicy -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasLoggingPolicy">
  <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
  <rdfs:range rdf:resource="&obliviouscloud;LoggingPolicy
      "/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasModificationApprovers -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasModificationApprovers">
  <rdfs:domain rdf:resource="&obliviouscloud;
      ModificationPolicy"/>
  <rdfs:range rdf:resource="&obliviouscloud;User"/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasModificationPolicy -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasModificationPolicy">
  <rdfs:range rdf:resource="&obliviouscloud;
      ModificationPolicy"/>
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
      "/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasPermissions -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasPermissions">
  <rdfs:range rdf:resource="&obliviouscloud;AccessLevel"/>
  <rdfs:domain rdf:resource="&obliviouscloud;Usergroup"/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasStoragePolicy -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;
      hasStoragePolicy">
  <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
  <rdfs:range rdf:resource="&obliviouscloud;StoragePolicy
      "/>
</owl:ObjectProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hasUsers -->
```

```
<owl:ObjectProperty rdf:about="&obliviouscloud;hasUsers">
  <rdfs:range rdf:resource="&obliviouscloud;User"/>
  <rdfs:domain rdf:resource="&obliviouscloud;Usergroup"/>
</owl:ObjectProperty>
```

```
<!--
```

```
////////////////////////////////////
```

```
//
// Data properties
//
////////////////////////////////////

—>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#approvalRequired —>
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      approvalRequired">
  <rdfs:domain rdf:resource="&obliviouscloud;
      ModificationPolicy"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#approversListInOrder —>
```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    approversListInOrder">
    <rdfs:domain rdf:resource="&obliviouscloud;
        ModificationPolicy"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#attributeBasedEncryption -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    attributeBasedEncryption">
    <rdfs:isDefinedBy rdf:datatype="&xsd:string">In public-
        key cryptography, a message is encrypted for a
        specific receiver using the receiver's public-key,
        in Attribute Based Encryption, identity can be set of
        values e.g. roles.

```

User should only be able to decrypt a ciphertext if the person holds a key for "matching attributes" (more below) where user keys are always issued by some trusted party.

```

<rdfs:domain rdf:resource="&obliviouscloud;
    EncryptionPolicy"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#auditPeriod -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    auditPeriod">
    <rdfs:domain rdf:resource="&obliviouscloud;AuditPolicy
        "/>
    <rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#author -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;author">
    <rdfs:domain rdf:resource="&obliviouscloud;Document"/>
    <rdfs:range rdf:resource="&xsd:string"/>

```

```
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#blockSize -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;blockSize
">
  <rdfs:isDefinedBy rdf:datatype="&xsd:string">Block size
    in KB</rdfs:isDefinedBy>
  <rdfs:domain rdf:resource="&obliviouscloud;
    ChunkingPolicy"/>
  <rdfs:range rdf:resource="&xsd:long"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#chunkCount -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
  chunkCount">
  <rdfs:domain rdf:resource="&obliviouscloud;
    ChunkingPolicy"/>
```



```

    <rdfs:range rdf:resource="&xsd;integer"/>
  </owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#chunkSize -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;chunkSize
    ">
    <rdfs:domain rdf:resource="&obliviouscloud;
        ChunkingPolicy"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
  </owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#chunksOnMultipleCloud -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    chunksOnMultipleCloud">
    <rdfs:domain rdf:resource="&obliviouscloud;
        ChunkingPolicy"/>
    <rdfs:range rdf:resource="&xsd;boolean"/>

```

```
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#cloudAuditor -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      cloudAuditor">
  <rdfs:domain rdf:resource="&obliviouscloud;AuditPolicy
      "/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#collectionDescription -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      collectionDescription">
  <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#collectionId -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      collectionId">
  <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
  <rdfs:range rdf:resource="&xsd;long"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#collectionName -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      collectionName">
  <rdfs:domain rdf:resource="&obliviouscloud;Collection"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#cryptoperiod -->

<owl:DatatypeProperty rdf:about="&obliviouscloud;
      cryptoperiod">
  <rdfs:isDefinedBy rdf:datatype="&xsd:string">The time in
      days span during which a specific key is authorized
      for use by legitimate
entities , or the keys for a given system will remain in
      effect </rdfs:isDefinedBy>
  <rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
      "/>
  <rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#docId -->

<owl:DatatypeProperty rdf:about="&obliviouscloud;docId">
  <rdfs:domain rdf:resource="&obliviouscloud;Document"/>
  <rdfs:range rdf:resource="&xsd:long"/>
</owl:DatatypeProperty>

```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#docTitle -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;docTitle
    ">
    <rdfs:domain rdf:resource="&obliviouscloud;Document"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#docUploadDate -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
    docUploadDate">
    <rdfs:domain rdf:resource="&obliviouscloud;Document"/>
    <rdfs:range rdf:resource="&xsd;dateTime"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#duplication -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      duplication">
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
      "/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#encryptionAlgorithm -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      encryptionAlgorithm">
  <rdfs:domain rdf:resource="&obliviouscloud;
      EncryptionPolicy"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#encryptionType -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      encryptionType">
  <rdfs:comment rdf:datatype="&xsd:string">example ,
      hardware encryption.</rdfs:comment>
  <rdfs:domain rdf:resource="&obliviouscloud;
      EncryptionPolicy"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hardwareBasedAuthenticator -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      hardwareBasedAuthenticator">
  <rdfs:domain rdf:resource="&obliviouscloud;
      EncryptionPolicy"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#hardwareBasedKeyStorage -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      hardwareBasedKeyStorage">
  <rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
    "/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#identityBasedEncryption -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      identityBasedEncryption">
  <rdfs:isDefinedBy rdf:datatype="&xsd:string">In public-
    key cryptography, a message is encrypted for a
    specific receiver using the receiver's public-key,
    in Identity Based Encryption, identity to be an
    arbitrary string, e.g., the email address of the
    receiver.</rdfs:isDefinedBy>
```



```

<rdfs:domain rdf:resource="&obliviouscloud;
    EncryptionPolicy"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#keyBackupProtocol -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    keyBackupProtocol">
    <rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
        "/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#keyDeregistration -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    keyDeregistration">

```

```

<rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#keyDestructionProtocol -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
keyDestructionProtocol">
<rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#keyLength -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;keyLength
">

```

```

<rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#keyRecoveryProtocol -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
keyRecoveryProtocol">
<rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#keyRegistration -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
keyRegistration">

```

```

<rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#keyRevocationProtocol -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
keyRevocationProtocol">
<rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#keyType -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;keyType">
<rdfs:comment rdf:datatype="&xsd:string">private , public
, symmentric , asymmentric etc .

```

```

[encryption key, authentication key, signature key, master
  key, transport key, agreement key, wrapping key]</rdfs:
  comment>

  <rdfs:domain rdf:resource="&obliviouscloud;KeyManagement
    "/>

  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
  securecloudstorage.owl#lastUpdated -->

<owl:DatatypeProperty rdf:about="&obliviouscloud;
  lastUpdated">
  <rdfs:domain rdf:resource="&obliviouscloud;Document"/>
  <rdfs:range rdf:resource="&xsd:dateTime"/>
</owl:DatatypeProperty>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
  securecloudstorage.owl#lastUpdatedBy -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    lastUpdatedBy">
    <rdfs:domain rdf:resource="&obliviouscloud;Document"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#loggingLevel -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    loggingLevel">
    <rdfs:domain rdf:resource="&obliviouscloud;LoggingPolicy
        "/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#loggingOnOperations -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    loggingOnOperations">

```

```

<rdfs:domain rdf:resource="&obliviouscloud;LoggingPolicy
"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#multiuserSSE -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
multiuserSSE">
<rdfs:domain rdf:resource="&obliviouscloud;
EncryptionPolicy"/>
<rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#password -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;password
">
<rdfs:domain rdf:resource="&obliviouscloud;User"/>

```

```

    <rdfs:range rdf:resource="&xsd:string"/>
  </owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#proofOfOwnership -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
      proofOfOwnership">
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
    "/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#proofOfRetrievability -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
      proofOfRetrievability">
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
    "/>
  <rdfs:range rdf:resource="&xsd:string"/>

```



```
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#proofOfStorage -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      proofOfStorage">
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
      "/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#proofOfStorageWithDeduplication
      -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      proofOfStorageWithDeduplication">
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
      "/>
  <rdfs:range rdf:resource="&xsd:string"/>
```

```
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#randomAudit -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      randomAudit">
  <rdfs:domain rdf:resource="&obliviouscloud;AuditPolicy
      "/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#redundancyCount -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      redundancyCount">
  <rdfs:domain rdf:resource="&obliviouscloud;StoragePolicy
      "/>
  <rdfs:range rdf:resource="&xsd:integer"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#searchableEncryption -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      searchableEncryption">
  <rdfs:domain rdf:resource="&obliviouscloud;
      EncryptionPolicy"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#sslCertHashingAlgorithm -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;
      sslCertHashingAlgorithm">
  <rdfs:domain rdf:resource="&obliviouscloud;
      EncryptionPolicy"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#userEmail -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;userEmail
    ">
    <rdfs:domain rdf:resource="&obliviouscloud;User"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#userId -->
```

```
<owl:DatatypeProperty rdf:about="&obliviouscloud;userId">
    <rdfs:domain rdf:resource="&obliviouscloud;User"/>
    <rdfs:range rdf:resource="&xsd:long"/>
</owl:DatatypeProperty>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#userName -->
```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;userName
    ">
    <rdfs:domain rdf:resource="&obliviouscloud;User"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#usergroupId -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    usergroupId">
    <rdfs:domain rdf:resource="&obliviouscloud;Usergroup"/>
    <rdfs:range rdf:resource="&xsd:long"/>
</owl:DatatypeProperty>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#usergroupName -->

```

```

<owl:DatatypeProperty rdf:about="&obliviouscloud;
    usergroupName">

```

```

    <rdfs:domain rdf:resource="&obliviouscloud;Usergroup"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```

<!--

```

```

////////////////////////////////////

```

```

//

```

```

// Classes

```

```

//

```

```

////////////////////////////////////

```

```

-->

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#ADMIN -->

```

```

<owl:Class rdf:about="&obliviouscloud;ADMIN">
  <rdfs:subClassOf rdf:resource="&obliviouscloud;
    AccessLevel"/>

```

```
</owl:Class>
```

```
<!-- http://userpages.umbc.edu/~vaishal1/ontologies/  
securecloudstorage.owl#AccessLevel -->
```

```
<owl:Class rdf:about="&obliviouscloud;AccessLevel"/>
```

```
<!-- http://userpages.umbc.edu/~vaishal1/ontologies/  
securecloudstorage.owl#AuditPolicy -->
```

```
<owl:Class rdf:about="&obliviouscloud;AuditPolicy"/>
```

```
<!-- http://userpages.umbc.edu/~vaishal1/ontologies/  
securecloudstorage.owl#ChunkingPolicy -->
```

```
<owl:Class rdf:about="&obliviouscloud;ChunkingPolicy">
```

```
<rdfs:subClassOf rdf:resource="&obliviouscloud;  
StoragePolicy"/>
```

```
</owl:Class>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#Collection -->
```

```
<owl:Class rdf:about="&obliviouscloud;Collection"/>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#DeletionPolicy -->
```

```
<owl:Class rdf:about="&obliviouscloud;DeletionPolicy">
```

```
  <rdfs:subClassOf rdf:resource="&obliviouscloud;
      StoragePolicy"/>
```

```
</owl:Class>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#Document -->
```

```
<owl:Class rdf:about="&obliviouscloud;Document"/>
```



```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#EncryptionPolicy -->
```

```
<owl:Class rdf:about="&obliviouscloud;EncryptionPolicy"/>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#KeyManagement -->
```

```
<owl:Class rdf:about="&obliviouscloud;KeyManagement">
  <rdfs:subClassOf rdf:resource="&obliviouscloud;
      EncryptionPolicy"/>
</owl:Class>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#LoggingPolicy -->
```

```
<owl:Class rdf:about="&obliviouscloud;LoggingPolicy">
  <rdfs:subClassOf rdf:resource="&obliviouscloud;
      StoragePolicy"/>
```

```
</owl:Class>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/  
securecloudstorage.owl#ModificationPolicy -->
```

```
<owl:Class rdf:about="&obliviouscloud;ModificationPolicy">  
  <rdfs:subClassOf rdf:resource="&obliviouscloud;  
    StoragePolicy"/>  
</owl:Class>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/  
securecloudstorage.owl#READ -->
```

```
<owl:Class rdf:about="&obliviouscloud;READ">  
  <rdfs:subClassOf rdf:resource="&obliviouscloud;  
    AccessLevel"/>  
</owl:Class>
```

```
<!-- http://userpages.umbc.edu/~vaishal1/ontologies /
      securecloudstorage.owl#StoragePolicy -->
```

```
<owl:Class rdf:about="&obliviouscloud;StoragePolicy"/>
```

```
<!-- http://userpages.umbc.edu/~vaishal1/ontologies /
      securecloudstorage.owl#User -->
```

```
<owl:Class rdf:about="&obliviouscloud;User"/>
```

```
<!-- http://userpages.umbc.edu/~vaishal1/ontologies /
      securecloudstorage.owl#Usergroup -->
```

```
<owl:Class rdf:about="&obliviouscloud;Usergroup"/>
```

```
<!-- http://userpages.umbc.edu/~vaishal1/ontologies /
      securecloudstorage.owl#WRITE -->
```

```
<owl:Class rdf:about="&obliviouscloud;WRITE">
```

```

    <rdfs:subClassOf rdf:resource="&obliviouscloud;
        AccessLevel"/>
</owl:Class>

<!-- http://userpages.umbc.edu/~vaishall/ontologies/
    securecloudstorage.owl#WhitelistedCloudProviders -->

<owl:Class rdf:about="&obliviouscloud;
    WhitelistedCloudProviders">
    <rdfs:comment rdf:datatype="&xsd:string">This is
        contains list of whitelisted cloud providers , that is
        only approved and trusted cloud providers that
        client wants to work with. If multi-cloud or vendor
        based solution is used then all cloud providers in
        the vendor solution should be whitelisted.</rdfs:
        comment>
</owl:Class>

<!-- http://userpages.umbc.edu/~vaishall/ontologies/
    securecloudstorage.owl#WhitelistedVendors -->

```

```

<owl:Class rdf:about="&obliviouscloud;WhitelistedVendors">
  <rdfs:comment rdf:datatype="&xsd:string">This is
    contains list of whitelisted vendors, that is only
    approved and trusted vendors that client wants to
    work with.</rdfs:comment>
</owl:Class>

```

```

<!--

```

```

////////////////////////////////////

```

```

//

```

```

// Individuals

```

```

//

```

```

////////////////////////////////////

```

```

-->

```

```

<!-- http://userpages.umbc.edu/~vaishall/ontologies/
    securecloudstorage.owl#Amazon -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;Amazon">
  <rdf:type rdf:resource="&obliviouscloud;
    WhitelistedCloudProviders"/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishal1/ontologies/
  securecloudstorage.owl#ArchivedAndInaccessible -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
  ArchivedAndInaccessible">
  <rdf:type rdf:resource="&obliviouscloud;DeletionPolicy
    "/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishal1/ontologies/
  securecloudstorage.owl#BiweeklyAudit -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
  BiweeklyAudit">
  <rdf:type rdf:resource="&obliviouscloud;AuditPolicy"/>

```

```

    <auditPeriod rdf:datatype="&xsd;integer">15</auditPeriod
  >
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#ClientSideKeySore -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
  ClientSideKeySore">
  <rdf:type rdf:resource="&obliviouscloud;KeyManagement"/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#Cloud9Vendor -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
  Cloud9Vendor">
  <rdf:type rdf:resource="&obliviouscloud;
    WhitelistedVendors"/>
  <hasCloudProviders rdf:resource="&obliviouscloud;Amazon
    "/>

```

```

    <hasCloudProviders rdf:resource="&obliviouscloud;
        MicrosoftAzure"/>
</owl:NamedIndividual>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#ComputeServerStorage -->

<owl:NamedIndividual rdf:about="&obliviouscloud;
    ComputeServerStorage">
    <rdf:type rdf:resource="&obliviouscloud;StoragePolicy"/>
    <redundancyCount rdf:datatype="&xsd;long">1</
        redundancyCount>
</owl:NamedIndividual>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#DEBUG -->

<owl:NamedIndividual rdf:about="&obliviouscloud;DEBUG">
    <rdf:type rdf:resource="&obliviouscloud;LoggingPolicy"/>
</owl:NamedIndividual>

```



```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#DeleteKeyOnly -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      DeleteKeyOnly">
  <rdf:type rdf:resource="&obliviouscloud;DeletionPolicy
      "/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#ERROR -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;ERROR">
  <rdf:type rdf:resource="&obliviouscloud;LoggingPolicy"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#EncryptedCloudKeyStore -->
```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    EncryptedCloudKeyStore">
    <rdf:type rdf:resource="&obliviouscloud;KeyManagement"/>
    <keyLength rdf:datatype="&xsd;integer">32</keyLength>
    <keyType rdf:datatype="&xsd;string">asymmetric </keyType>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#ExtraSecureKeyStore -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    ExtraSecureKeyStore">
    <rdf:type rdf:resource="&obliviouscloud;KeyManagement"/>
    <keyLength rdf:datatype="&xsd;integer">64</keyLength>
    <keyType rdf:datatype="&xsd;string">symmentric </keyType>
    <hardwareBasedKeyStorage rdf:datatype="&xsd;boolean">
        true </hardwareBasedKeyStorage>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#HR -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;HR">
  <rdf:type rdf:resource="&obliviouscloud;Usergroup"/>
  <usergroupName rdf:datatype="&xsd:string">HR</
    usergroupName>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishall/ontologies/
  securecloudstorage.owl#Lenient_EP -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;Lenient_EP
">
  <rdf:type rdf:resource="&obliviouscloud;EncryptionPolicy
"/>
  <encryptionAlgorithm rdf:datatype="&xsd:string">Tiny
    Encryption Algo</encryptionAlgorithm>
  <searchableEncryption rdf:datatype="&xsd:boolean">true </
    searchableEncryption>
  <hasKeyManagement rdf:resource="&obliviouscloud;
    ClientSideKeyStore"/>
</owl:NamedIndividual>

```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#MicrosoftAzure -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      MicrosoftAzure">
  <rdf:type rdf:resource="&obliviouscloud;
      WhitelistedCloudProviders"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#ModifyWithApproval -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      ModifyWithApproval">
  <rdf:type rdf:resource="&obliviouscloud;
      ModificationPolicy"/>
  <approvalRequired rdf:datatype="&xsd:boolean">true </
      approvalRequired>
  <hasModificationApprovers rdf:resource="&obliviouscloud;
      UserGroup1"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#MonthlyAudit -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      MonthlyAudit">
  <rdf:type rdf:resource="&obliviouscloud;AuditPolicy"/>
  <auditPeriod rdf:datatype="&xsd;integer">30</auditPeriod
    >
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#NoChunking -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;NoChunking
      ">
  <rdf:type rdf:resource="&obliviouscloud;ChunkingPolicy
      "/>
  <chunkCount rdf:datatype="&xsd;integer">1</chunkCount>
</owl:NamedIndividual>
```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#RSA_moderate_EP -->

<owl:NamedIndividual rdf:about="&obliviouscloud;
      RSA_moderate_EP">
  <rdf:type rdf:resource="&obliviouscloud;EncryptionPolicy
    "/>
  <attributeBasedEncryption rdf:datatype="&xsd:string">NA
    </attributeBasedEncryption>
  <encryptionAlgorithm rdf:datatype="&xsd:string">RSA</
    encryptionAlgorithm>
  <encryptionType rdf:datatype="&xsd:string">asymmetric</
    encryptionType>
  <identityBasedEncryption rdf:datatype="&xsd:string">
    email</identityBasedEncryption>
  <searchableEncryption rdf:datatype="&xsd:boolean">>false
    </searchableEncryption>
  <hasKeyManagement rdf:resource="&obliviouscloud;
    EncryptedCloudKeyStore"/>
</owl:NamedIndividual>

```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#RandomAudit -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
RandomAudit">
  <rdf:type rdf:resource="&obliviouscloud;AuditPolicy"/>
  <randomAudit rdf:datatype="&xsd:boolean">true </
randomAudit>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#SecureWipe -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;SecureWipe
">
  <rdf:type rdf:resource="&obliviouscloud;DeletionPolicy
"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
securecloudstorage.owl#SimpleDataDeletion -->
```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    SimpleDataDeletion">
    <rdf:type rdf:resource="&obliviouscloud;DeletionPolicy
        "/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#SimpleStorage2 -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    SimpleStorage2">
    <rdf:type rdf:resource="&obliviouscloud;StoragePolicy"/>
    <redundancyCount rdf:datatype="&xsd;integer">2</
        redundancyCount>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#SimplyModify -->

```



```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    SimplyModify">
    <rdf:type rdf:resource="&obliviouscloud;
        ModificationPolicy"/>
    <approvalRequired rdf:datatype="&xsd:boolean">false </
        approvalRequired>
</owl:NamedIndividual>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#Strict_EP -->

<owl:NamedIndividual rdf:about="&obliviouscloud; Strict_EP
    ">
    <rdf:type rdf:resource="&obliviouscloud; EncryptionPolicy
        "/>
    <encryptionAlgorithm rdf:datatype="&xsd:string">RSA</
        encryptionAlgorithm>
    <encryptionType rdf:datatype="&xsd:string">Symmetric</
        encryptionType>
    <multiuserSSE rdf:datatype="&xsd:boolean">true </
        multiuserSSE>
    <hasKeyManagement rdf:resource="&obliviouscloud;
        ExtraSecureKeyStore"/>

```

```
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#SuperVisorGroup1 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      SuperVisorGroup1">
  <rdf:type rdf:resource="&obliviouscloud;Usergroup"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#Supervisor1 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      Supervisor1">
  <rdf:type rdf:resource="&obliviouscloud;User"/>
  <userEmail rdf:datatype="&xsd:string">supervisor1@onr.
      com</userEmail>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#User1 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;User1">
  <rdf:type rdf:resource="&obliviouscloud;User"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#User2 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;User2">
  <rdf:type rdf:resource="&obliviouscloud;User"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#UserGroup1 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;UserGroup1
">
  <rdf:type rdf:resource="&obliviouscloud;Usergroup"/>
```

```

    <hasUsers rdf:resource="&obliviouscloud;User1"/>
    <hasUsers rdf:resource="&obliviouscloud;User2"/>
    <hasAccessToCollections rdf:resource="&obliviouscloud;cl
        "/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#UserHelpCollection -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    UserHelpCollection">
    <rdf:type rdf:resource="&obliviouscloud;Collection"/>
    <collectionId rdf:datatype="&xsd;long">99999</
        collectionId>
    <collectionDescription rdf:datatype="&xsd;string">This
        collection holds user help documents.</
        collectionDescription>
    <collectionName rdf:datatype="&xsd;string">
        UserHelpCollection </collectionName>
    <hasDeletionPolicy rdf:resource="&obliviouscloud;
        DeleteKeyOnly"/>
    <hasLoggingPolicy rdf:resource="&obliviouscloud;ERROR"/>

```

```

<hasEncryptionPolicy rdf:resource="&obliviouscloud;
    RSA_moderate_EP"/>
<hasAuditPolicy rdf:resource="&obliviouscloud;
    RandomAudit"/>
<hasStoragePolicy rdf:resource="&obliviouscloud;
    SimpleStorage2"/>
<hasDocuments rdf:resource="&obliviouscloud;UserHelpDoc1
    "/>
<hasDocuments rdf:resource="&obliviouscloud;UserHelpDoc2
    "/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#UserHelpDoc1 -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    UserHelpDoc1">
    <rdf:type rdf:resource="&obliviouscloud;Document"/>
</owl:NamedIndividual>

```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#UserHelpDoc2 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      UserHelpDoc2">
  <rdf:type rdf:resource="&obliviouscloud;Document"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#WARN -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;WARN">
  <rdf:type rdf:resource="&obliviouscloud;LoggingPolicy"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#WeeklyAudit -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      WeeklyAudit">
  <rdf:type rdf:resource="&obliviouscloud;AuditPolicy"/>
```

```

    <auditPeriod rdf:datatype="&xsd;integer">7</auditPeriod>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#adminAccess -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    adminAccess">
    <rdf:type rdf:resource="&obliviouscloud;AccessLevel"/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#adminGroup -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;adminGroup
    ">
    <rdf:type rdf:resource="&obliviouscloud;Usergroup"/>
    <usergroupId rdf:datatype="&xsd;long">1</usergroupId>
    <usergroupName rdf:datatype="&xsd;string">adminGroup</
        usergroupName>
</owl:NamedIndividual>

```

```
<!-- http://userpages.umbc.edu/~vaishall/ontologies/
      securecloudstorage.owl#c1 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;c1">
  <rdf:type rdf:resource="&obliviouscloud;Collection"/>
  <collectionId rdf:datatype="&xsd;long">123456</
    collectionId>
  <collectionDescription rdf:datatype="&xsd;string">This
    is sample collection </collectionDescription>
  <collectionName rdf:datatype="&xsd;string">
    sampleCollection </collectionName>
  <hasAuditPolicy rdf:resource="&obliviouscloud;
    BiweeklyAudit"/>
  <hasStoragePolicy rdf:resource="&obliviouscloud;
    ComputeServerStorage"/>
  <hasEncryptionPolicy rdf:resource="&obliviouscloud;
    RSA_moderate_EP"/>
  <hasDeletionPolicy rdf:resource="&obliviouscloud;
    SecureWipe"/>
  <hasLoggingPolicy rdf:resource="&obliviouscloud;WARN"/>
  <hasDocuments rdf:resource="&obliviouscloud;d1"/>
</owl:NamedIndividual>
```



```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#d1 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;d1">
  <rdf:type rdf:resource="&obliviouscloud;Document"/>
  <docId rdf:datatype="&xsd;long">12345</docId>
  <docUploadDate rdf:datatype="&xsd;dateTime">2014-09-23
    T23:03:11Z</docUploadDate>
  <author rdf:datatype="&xsd:string">ABC</author>
  <docTitle rdf:datatype="&xsd:string">Sample Document D1
    </docTitle>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#modifyAccess -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      modifyAccess">
  <rdf:type rdf:resource="&obliviouscloud;AccessLevel"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#readAccess -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;readAccess
">
  <rdf:type rdf:resource="&obliviouscloud;AccessLevel"/>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#rootUser -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;rootUser">
  <rdf:type rdf:resource="&obliviouscloud;User"/>
  <userId rdf:datatype="&xsd;long">1</userId>
  <userEmail rdf:datatype="&xsd:string">root@onr.com</
    userEmail>
  <userName rdf:datatype="&xsd:string">rootUser </userName>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#splitIn256kb -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;
      splitIn256kb">
  <rdf:type rdf:resource="&obliviouscloud;ChunkingPolicy
      "/>
  <chunkSize rdf:datatype="&xsd;integer">256</chunkSize>
  <chunksOnMultipleCloud rdf:datatype="&xsd;boolean">true
      </chunksOnMultipleCloud>
</owl:NamedIndividual>
```

```
<!-- http://userpages.umbc.edu/~vaishali/ontologies/
      securecloudstorage.owl#splitIn3 -->
```

```
<owl:NamedIndividual rdf:about="&obliviouscloud;splitIn3">
  <rdf:type rdf:resource="&obliviouscloud;ChunkingPolicy
      "/>
  <chunkCount rdf:datatype="&xsd;integer">3</chunkCount>
  <hasChunksOnClouds rdf:resource="&obliviouscloud;Amazon
      "/>
```

```

    <hasChunksOnClouds rdf:resource="&obliviouscloud;
        MicrosoftAzure"/>
</owl:NamedIndividual>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#superCollection -->

<owl:NamedIndividual rdf:about="&obliviouscloud;
    superCollection">
    <rdf:type rdf:resource="&obliviouscloud;Collection"/>
    <collectionId rdf:datatype="&xsd;long">1</collectionId>
    <collectionDescription rdf:datatype="&xsd;string">This
        is System Generated root level collection.</
        collectionDescription>
    <collectionName rdf:datatype="&xsd;string">
        superCollection </collectionName>
    <hasDocuments rdf:resource="&obliviouscloud;d1"/>
</owl:NamedIndividual>

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#superUserGroup -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    superUserGroup">
    <rdf:type rdf:resource="&obliviouscloud;Usergroup"/>
    <usergroupId rdf:datatype="&xsd;long">1</usergroupId>
    <usergroupName rdf:datatype="&xsd;string">superUserGroup
        </usergroupName>
    <hasUsers rdf:resource="&obliviouscloud;rootUser"/>
    <hasAccessToCollections rdf:resource="&obliviouscloud;
        superCollection"/>
</owl:NamedIndividual>

```

```

<!-- http://userpages.umbc.edu/~vaishali/ontologies/
    securecloudstorage.owl#test_class_2_inst -->

```

```

<owl:NamedIndividual rdf:about="&obliviouscloud;
    test_class_2_inst"/>

```

```

<!--

```

```

////////////////////////////////////

```

```
//
```

```
// Rules
```

```
//
```

```
////////////////////////////////////
```

```
—>
```

```
<rdf:Description rdf:about="urn:swrl#x">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
<rdf:Description rdf:about="urn:swrl#y">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
<rdf:Description rdf:about="urn:swrl#ep">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
<rdf:Description rdf:about="urn:swrl#s">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
<rdf:Description rdf:about="urn:swrl#z">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
<rdf:Description rdf:about="urn:swrl#et">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
```

```

<rdf:Description rdf:about="urn:swrl#mp">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
<rdf:Description rdf:about="urn:swrl#ar">
  <rdf:type rdf:resource="&swrl;Variable"/>
</rdf:Description>
<rdf:Description>
  <rdf:type rdf:resource="&swrl;Imp"/>
  <swrl:head>
    <rdf:Description>
      <rdf:type rdf:resource="&swrl;AtomList"/>
      <rdf:rest rdf:resource="&rdf:nil"/>
      <rdf:first>
        <rdf:Description>
          <rdf:type rdf:resource="&swrl;
            DatavaluedPropertyAtom"/>
          <swrl:argument2 rdf:datatype="&xsd;integer">-1</
            swrl:argument2>
          <swrl:propertyPredicate rdf:resource="&
            obliviouscloud;chunkSize"/>
          <swrl:argument1 rdf:resource="urn:swrl#x"/>
        </rdf:Description>
      </rdf:first>
    </rdf:Description>
  </swrl:head>

```

```

<swrl:body>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:first>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;ClassAtom"/>
        <swrl:classPredicate rdf:resource="&
          obliviouscloud;ChunkingPolicy"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
      </rdf:Description>
    </rdf:first>
    <rdf:rest>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;AtomList"/>
        <rdf:first>
          <rdf:Description>
            <rdf:type rdf:resource="&swrl;DataRangeAtom
              "/>
            <swrl:argument1 rdf:resource="urn:swrl#y"/>
            <swrl:dataRange>
              <rdfs:Datatype>
                <owl:onDatatype rdf:resource="&xsd;
                  integer"/>
                <owl:withRestrictions rdf:parseType="
                  Collection">

```



```

    <rdf:Description>
      <xsd:minExclusive rdf:datatype="&xsd
        ;integer">0</xsd:minExclusive>
    </rdf:Description>
  </owl:withRestrictions>
</rdfs:Datatype>
</swrl:dataRange>
</rdf:Description>
</rdf:first>
<rdf:rest>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest rdf:resource="&rdf:nil"/>
    <rdf:first>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;
          DatavaluedPropertyAtom"/>
        <swrl:propertyPredicate rdf:resource="&
          obliviouscloud;chunkCount"/>
        <swrl:argument1 rdf:resource="urn:swrl#x
          "/>
        <swrl:argument2 rdf:resource="urn:swrl#y
          "/>
      </rdf:Description>
    </rdf:first>
  </rdf:rest>
</rdf:Description>

```

```

        </rdf:Description>
    </rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</swrl:body>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&swrl;Imp"/>
    <swrl:body>
        <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>
            <rdf:first>
                <rdf:Description>
                    <rdf:type rdf:resource="&swrl;ClassAtom"/>
                    <swrl:classPredicate rdf:resource="&obliviouscloud;ChunkingPolicy"/>
                    <swrl:argument1 rdf:resource="urn:swrl#x"/>
                </rdf:Description>
            </rdf:first>
            <rdf:rest>
                <rdf:Description>
                    <rdf:type rdf:resource="&swrl;AtomList"/>
                    <rdf:rest rdf:resource="&rdf:nil"/>
                    <rdf:first>

```

```

<rdf:Description>
  <rdf:type rdf:resource="&swrl;
    DatavaluedPropertyAtom"/>
  <swrl:argument2 rdf:datatype="&xsd;integer
    ">1</swrl:argument2>
  <swrl:propertyPredicate rdf:resource="&
    obliviouscloud;chunkCount"/>
  <swrl:argument1 rdf:resource="urn:swrl#x"/>
</rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</swrl:body>
<swrl:head>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest rdf:resource="&rdf:nil"/>
    <rdf:first>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;
          DatavaluedPropertyAtom"/>
        <swrl:argument2 rdf:datatype="&xsd;boolean">
          false </swrl:argument2>

```

```

        <swrl:propertyPredicate rdf:resource="&
            obliviouscloud;chunksOnMultipleCloud"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:head>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&swrl;Imp"/>
    <swrl:body>
        <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>
            <rdf:rest>
                <rdf:Description>
                    <rdf:type rdf:resource="&swrl;AtomList"/>
                    <rdf:rest rdf:resource="&rdf:nil"/>
                    <rdf:first>
                        <rdf:Description>
                            <rdf:type rdf:resource="&swrl;
                                DatavaluedPropertyAtom"/>
                            <swrl:argument2 rdf:datatype="&xsd;boolean">
                                true </swrl:argument2>
                            <swrl:propertyPredicate rdf:resource="&
                                obliviouscloud;chunksOnMultipleCloud"/>

```

```

        <swrl:argument1 rdf:resource="urn:swrl#x"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
<rdf:first>
    <rdf:Description>
        <rdf:type rdf:resource="&swrl;ClassAtom"/>
        <swrl:classPredicate rdf:resource="&
            obliviouscloud;ChunkingPolicy"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:body>
<swrl:head>
    <rdf:Description>
        <rdf:type rdf:resource="&swrl;AtomList"/>
        <rdf:rest rdf:resource="&rdf:nil"/>
        <rdf:first>
            <rdf:Description>
                <rdf:type rdf:resource="&swrl;
                    IndividualPropertyAtom"/>
                <swrl:argument2 rdf:resource="&obliviouscloud;
                    Amazon"/>

```

```

        <swrl:propertyPredicate rdf:resource="&
            obliviouscloud;hasChunksOnClouds"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:head>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&swrl;Imp"/>
    <swrl:head>
        <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>
            <rdf:rest rdf:resource="&rdf:nil"/>
            <rdf:first>
                <rdf:Description>
                    <rdf:type rdf:resource="&swrl;
                        IndividualPropertyAtom"/>
                    <swrl:argument2 rdf:resource="&obliviouscloud;
                        SuperVisorGroup1"/>
                    <swrl:propertyPredicate rdf:resource="&
                        obliviouscloud;hasModificationApprovers"/>
                    <swrl:argument1 rdf:resource="urn:swrl#mp"/>
                </rdf:Description>
            </rdf:first>

```

```

    </rdf:Description>
</swrl:head>
<swrl:body>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:first>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;ClassAtom"/>
        <swrl:classPredicate rdf:resource="&
          obliviouscloud;ModificationPolicy"/>
        <swrl:argument1 rdf:resource="urn:swrl#mp"/>
      </rdf:Description>
    </rdf:first>
    <rdf:rest>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;AtomList"/>
        <rdf:rest>
          <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>
            <rdf:rest rdf:resource="&rdf:nil"/>
            <rdf:first>
              <rdf:Description>
                <rdf:type rdf:resource="&swrl;
                  BuiltinAtom"/>

```

```

<swrl:builtin rdf:resource="&swrlb;equal
"/>
<swrl:arguments>
  <rdf:Description>
    <rdf:type rdf:resource="&rdf;List"/>
    <rdf:first rdf:resource="urn:swrl#ar
"/>
  <rdf:rest>
    <rdf:Description>
      <rdf:type rdf:resource="&rdf;
List"/>
      <rdf:first rdf:datatype="&xsd;
boolean">true </rdf:first>
      <rdf:rest rdf:resource="&rdf:nil
"/>
    </rdf:Description>
  </rdf:rest>
</rdf:Description>
</swrl:arguments>
</rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
<rdf:first>
  <rdf:Description>

```



```

        <rdf:type rdf:resource="&swrl;
            DatavaluedPropertyAtom"/>
        <swrl:propertyPredicate rdf:resource="&
            obliviouscloud;approvalRequired"/>
        <swrl:argument2 rdf:resource="urn:swrl#ar"/>
        <swrl:argument1 rdf:resource="urn:swrl#mp"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</swrl:body>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&swrl;Imp"/>
    <swrl:body>
        <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>
            <rdf:first>
                <rdf:Description>
                    <rdf:type rdf:resource="&swrl;ClassAtom"/>
                    <swrl:classPredicate rdf:resource="&
                        obliviouscloud;EncryptionPolicy"/>
                    <swrl:argument1 rdf:resource="urn:swrl#ep"/>
                </rdf:Description>

```

```

</rdf:first>
<rdf:rest>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:first>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;
          DatavaluedPropertyAtom"/>
        <swrl:propertyPredicate rdf:resource="&
          obliviouscloud;encryptionType"/>
        <swrl:argument1 rdf:resource="urn:swrl#ep"/>
        <swrl:argument2 rdf:resource="urn:swrl#et"/>
      </rdf:Description>
    </rdf:first>
  <rdf:rest>
    <rdf:Description>
      <rdf:type rdf:resource="&swrl;AtomList"/>
      <rdf:rest rdf:resource="&rdf:nil"/>
      <rdf:first>
        <rdf:Description>
          <rdf:type rdf:resource="&swrl;
            BuiltinAtom"/>
          <swrl:builtin rdf:resource="&swrlb;equal
            "/>
          <swrl:arguments>

```

```

<rdf:Description>
  <rdf:type rdf:resource="&rdf;List"/>
  <rdf:first rdf:resource="urn:swrl#et
    "/>
  <rdf:rest>
    <rdf:Description>
      <rdf:type rdf:resource="&rdf;
        List"/>
      <rdf:first>symmetric </rdf:first>
      <rdf:rest rdf:resource="&rdf;nil
        "/>
    </rdf:Description>
  </rdf:rest>
</rdf:Description>
</swrl:arguments>
</rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</swrl:body>
<swrl:head>
  <rdf:Description>

```

```

<rdf:type rdf:resource="&swrl;AtomList"/>
<rdf:rest rdf:resource="&rdf:nil"/>
<rdf:first>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;
      DatavaluedPropertyAtom"/>
    <swrl:argument2 rdf:datatype="&xsd:boolean">true
    </swrl:argument2>
    <swrl:propertyPredicate rdf:resource="&
      obliviouscloud;searchableEncryption"/>
    <swrl:argument1 rdf:resource="urn:swrl#ep"/>
  </rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:head>
</rdf:Description>
<rdf:Description>
  <rdf:type rdf:resource="&swrl;Imp"/>
  <swrl:head>
    <rdf:Description>
      <rdf:type rdf:resource="&swrl;AtomList"/>
      <rdf:rest rdf:resource="&rdf:nil"/>
      <rdf:first>
        <rdf:Description>

```

```

<rdf:type rdf:resource="&swrl;
    DatavaluedPropertyAtom"/>
<swrl:argument2 rdf:datatype="&xsd:boolean">true
    </swrl:argument2>
<swrl:propertyPredicate rdf:resource="&
    obliviouscloud;duplication"/>
<swrl:argument1 rdf:resource="urn:swrl#x"/>
</rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:head>
<swrl:body>
<rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest>
        <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>
            <rdf:rest>
                <rdf:Description>
                    <rdf:type rdf:resource="&swrl;AtomList"/>
                    <rdf:rest rdf:resource="&rdf:nil"/>
                    <rdf:first>
                        <rdf:Description>
                            <rdf:type rdf:resource="&swrl;
                                DataRangeAtom"/>

```

```

    <swrl:argument1 rdf:resource="urn:swrl#z
    "/>
    <swrl:dataRange>
      <rdfs:Datatype>
        <owl:onDatatype rdf:resource="&xsd;
        integer"/>
        <owl:withRestrictions rdf:parseType
        ="Collection">
          <rdf:Description>
            <xsd:minExclusive rdf:datatype
            ="&xsd;integer">1</xsd:
            minExclusive>
          </rdf:Description>
        </owl:withRestrictions>
      </rdfs:Datatype>
    </swrl:dataRange>
  </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
<rdf:first>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;
    DatavaluedPropertyAtom"/>

```

```

        <swrl:propertyPredicate rdf:resource="&
            obliviouscloud;redundancyCount"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
        <swrl:argument2 rdf:resource="urn:swrl#z"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
<rdf:first>
    <rdf:Description>
        <rdf:type rdf:resource="&swrl;ClassAtom"/>
        <swrl:classPredicate rdf:resource="&
            obliviouscloud;StoragePolicy"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:body>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&swrl;Imp"/>
    <swrl:head>
        <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>
            <rdf:rest rdf:resource="&rdf:nil"/>

```

```

<rdf: first >
  <rdf: Description >
    <rdf: type  rdf: resource="&swrl;
      DatavaluedPropertyAtom"/>
    <swrl: argument2>symmetric </swrl: argument2>
    <swrl: propertyPredicate  rdf: resource="&
      obliviouscloud; encryptionType"/>
    <swrl: argument1  rdf: resource="urn: swrl#ep"/>
  </rdf: Description >
</rdf: first >
</rdf: Description >
</swrl: head>
<swrl: body>
  <rdf: Description >
    <rdf: type  rdf: resource="&swrl; AtomList"/>
    <rdf: first >
      <rdf: Description >
        <rdf: type  rdf: resource="&swrl; ClassAtom"/>
        <swrl: classPredicate  rdf: resource="&
          obliviouscloud; EncryptionPolicy"/>
        <swrl: argument1  rdf: resource="urn: swrl#ep"/>
      </rdf: Description >
    </rdf: first >
    <rdf: rest >
      <rdf: Description >

```



```

<rdf:type rdf:resource="&swrl;AtomList"/>
<rdf:rest>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest rdf:resource="&rdf:nil"/>
    <rdf:first>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;
          BuiltinAtom"/>
        <swrl:builtin rdf:resource="&swrlb;equal
          "/>
        <swrl:arguments>
          <rdf:Description>
            <rdf:type rdf:resource="&rdf;List"/>
            <rdf:first rdf:resource="urn:swrl#s
              "/>
            <rdf:rest>
              <rdf:Description>
                <rdf:type rdf:resource="&rdf;
                  List"/>
                <rdf:first rdf:datatype="&xsd;
                  boolean">true </rdf:first>
                <rdf:rest rdf:resource="&rdf:nil
                  "/>
              </rdf:Description>

```

```

        </rdf:rest>
    </rdf:Description>
</swrl:arguments>
</rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
<rdf:first>
    <rdf:Description>
        <rdf:type rdf:resource="&swrl;
            DatavaluedPropertyAtom"/>
        <swrl:propertyPredicate rdf:resource="&
            obliviouscloud;searchableEncryption"/>
        <swrl:argument1 rdf:resource="urn:swrl#ep"/>
        <swrl:argument2 rdf:resource="urn:swrl#s"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</swrl:body>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&swrl;Imp"/>
    <swrl:head>

```

```

<rdf:Description>
  <rdf:type rdf:resource="&swrl;AtomList"/>
  <rdf:rest rdf:resource="&rdf:nil"/>
  <rdf:first>
    <rdf:Description>
      <rdf:type rdf:resource="&swrl;
        DatavaluedPropertyAtom"/>
      <swrl:argument2 rdf:datatype="&xsd:boolean">
        false </swrl:argument2>
      <swrl:propertyPredicate rdf:resource="&
        obliviouscloud;duplication"/>
      <swrl:argument1 rdf:resource="urn:swrl#x"/>
    </rdf:Description>
  </rdf:first>
</rdf:Description>
</swrl:head>
<swrl:body>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;AtomList"/>
        <rdf:rest>
          <rdf:Description>
            <rdf:type rdf:resource="&swrl;AtomList"/>

```

```

<rdf:rest rdf:resource="&rdf:nil"/>
<rdf:first>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;
      DataRangeAtom"/>
    <swrl:argument1 rdf:resource="urn:swrl#y
      "/>
    <swrl:dataRange>
      <rdfs:Datatype>
        <owl:onDatatype rdf:resource="&xsd;
          integer"/>
        <owl:withRestrictions rdf:parseType
          ="Collection">
          <rdf:Description>
            <xsd:maxInclusive rdf:datatype
              ="&xsd;integer">1</xsd:
              maxInclusive>
          </rdf:Description>
        </owl:withRestrictions>
      </rdfs:Datatype>
    </swrl:dataRange>
  </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>

```

```

<rdf:first>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;
      DatavaluedPropertyAtom"/>
    <swrl:propertyPredicate rdf:resource="&
      obliviouscloud;redundancyCount"/>
    <swrl:argument1 rdf:resource="urn:swrl#x"/>
    <swrl:argument2 rdf:resource="urn:swrl#y"/>
  </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
<rdf:first>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;ClassAtom"/>
    <swrl:classPredicate rdf:resource="&
      obliviouscloud;StoragePolicy"/>
    <swrl:argument1 rdf:resource="urn:swrl#x"/>
  </rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:body>
</rdf:Description>
<rdf:Description>
  <rdf:type rdf:resource="&swrl;Imp"/>

```

```

<swrl:head>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest rdf:resource="&rdf:nil"/>
    <rdf:first>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;
          DatavaluedPropertyAtom"/>
        <swrl:argument2 rdf:datatype="&xsd;integer">-1</
          swrl:argument2>
        <swrl:propertyPredicate rdf:resource="&
          obliviouscloud;chunkCount"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
      </rdf:Description>
    </rdf:first>
  </rdf:Description>
</swrl:head>
<swrl:body>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest>
      <rdf:Description>
        <rdf:type rdf:resource="&swrl;AtomList"/>
        <rdf:first>
          <rdf:Description>

```

```

<rdf:type rdf:resource="&swrl;DataRangeAtom
"/>
<swrl:argument1 rdf:resource="urn:swrl#y"/>
<swrl:dataRange>
  <rdfs:Datatype>
    <owl:onDatatype rdf:resource="&xsd;
      integer"/>
    <owl:withRestrictions rdf:parseType="
      Collection">
      <rdf:Description>
        <xsd:minExclusive rdf:datatype="&xsd
          ;integer">0</xsd:minExclusive>
      </rdf:Description>
    </owl:withRestrictions>
  </rdfs:Datatype>
</swrl:dataRange>
</rdf:Description>
</rdf:first>
<rdf:rest>
  <rdf:Description>
    <rdf:type rdf:resource="&swrl;AtomList"/>
    <rdf:rest rdf:resource="&rdf:nil"/>
    <rdf:first>
      <rdf:Description>

```

```

        <rdf:type rdf:resource="&swrl;
            DatavaluedPropertyAtom"/>
        <swrl:propertyPredicate rdf:resource="&
            obliviouscloud;chunkSize"/>
        <swrl:argument1 rdf:resource="urn:swrl#x
            "/>
        <swrl:argument2 rdf:resource="urn:swrl#y
            "/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
<rdf:first>
    <rdf:Description>
        <rdf:type rdf:resource="&swrl;ClassAtom"/>
        <swrl:classPredicate rdf:resource="&
            obliviouscloud;ChunkingPolicy"/>
        <swrl:argument1 rdf:resource="urn:swrl#x"/>
    </rdf:Description>
</rdf:first>
</rdf:Description>
</swrl:body>
</rdf:Description>

```



```
</rdf:RDF>
```

```
<!-- Generated by the OWL API (version 3.5.1) http://owlapi.  
sourceforge.net -->
```

REFERENCES

- [1] Oded Goldreich and Rafail Ostrovsky. *Software protection and simulation on oblivious rams*. J. ACM, 43(3):431473, 1996.
- [2] Ivan Damgard, Sigurd Meldgaard, and Jesper Buus Nielsen. *Perfectly secure oblivious RAM without random oracles*. In Yuval Ishai, editor, TCC 2011, volume 6597 of LNCS, pages 144163. Springer, March 2011.
- [3] Michael T. Goodrich, Michael Mitzenmacher, Olga Ohrimenko, and Roberto Tamassia. *Privacy-preserving group data access via stateless oblivious RAM simulation*. In Yuval Rabani, editor, 23rd SODA, pages 157167. ACM-SIAM, January 2012.
- [4] Eyal Kushilevitz, Steve Lu, and Rafail Ostrovsky. *On the (in)security of hash-based oblivious RAM and a new balancing scheme*. In Yuval Rabani, editor, 23rd SODA, pages 143156. ACM-SIAM, January 2012.
- [5] Emil Stefanov, Elaine Shi, and Dawn Xiaodong Song. *Towards practical oblivious ram*. In NDSS, 2012.
- [6] Emil Stefanov, Marten van Dijk, Elaine Shi, Christopher W. Fletcher, Ling Ren, Xiangyao Yu, and Srinivas Devadas. *Path ORAM: an extremely simple oblivious RAM protocol*. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, ACM CCS 13, pages 299310. ACM Press, November 2013.
- [7] Jonathan L. Dautrich Jr., Emil Stefanov, and Elaine Shi. *Burst ORAM: minimizing ORAM response times for bursty access patterns*. In Proceedings of the 23rd USENIX Security Symposium, pages 749764, 2014.

- [8] Xiao Shaun Wang, Kartik Nayak, Chang Liu, T.-H. Hubert Chan, Elaine Shi, Emil Stefanov, and Yan Huang. *Oblivious data structures*. In Gail- Joon Ahn, Moti Yung, and Ninghui Li, editors, ACM CCS 14, pages 215–226. ACM Press, November 2014.
- [9] Qingji Zheng, Shouhuai Xu, *Secure and Efficient Proof of Storage with Deduplication*, Proceedings of the second ACM conference on Data and Application Security and Privacy.
- [10] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan. *Sedic: privacyaware data intensive computing on hybrid clouds*. In CCS, 2011.
- [11] E. Stefanov, M. van Dijk, A. Oprea, and A. Juels. *Iris: A scalable cloud file system with efficient integrity checks*. In ACSAC, 2012.
- [12] P. Williams, R. Sion, and A. Tomescu. *Privatefs: A parallel oblivious file system*. In CCS, 2012.
- [13] Emil Stefanov, Elaine Shi, *ObliviStore: High Performance Oblivious Cloud Storage*, In IEEE Symposium on security and Privacy (SP), 2013.
- [14] *NIST Cloud Computing Security Reference Architecture*, NIST Special Publication 500-299.
- [15] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid, *Recommendation for Key Management Part 1: General (Revision 3)*, 2012.
- [16] Seny Kamara, Kristin Lauter, *Cryptographic Cloud Storage*, In Proceedings of the 14th international conference on Financial cryptography and data security, 2010.
- [17] *NIST Cloud Computing Use Cases, Section 3 Cloud Management Use cases, subsection 3.6. Erase Data Objects In a Cloud*.

- [18] D. McGuinness, F. Van Harmelen, et al., *OWL web ontology language overview*, W3C recommendation, World Wide Web Consortium, 2004.
- [19] A. Hendre, T. Finin, K.Joshi, *Cloud Security and Compliance Ontology*, July 2014, <http://ebiquity.umbc.edu/resource/html/id/361/Cloud-Security-and-Compliance-Ontology>.
- [20] Karuna Pande Joshi et al., *Automating Cloud Service Level Agreements using Semantic Technologies*, InProceedings, CLaw Workshop, IEEE International Conference on Cloud Engineering (IC2E), March 2015.
- [21] Karuna Pande Joshi et al., *Automating Cloud Services Lifecycle through Semantic technologies*, Article, IEEE Transactions on Service Computing, January 2014.
- [22] Natalya F. Noy and Deborah L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*
- [23] Cross-site Request Forgery [https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF))
- [24] Julita Bermejo, *A Simplified Guide to Create an Ontology*, ASLab R- 2007-004 v 0.1 Draft May 22, 2007.
- [25] Cloud Security Alliance, *Security Guidance for Critical Areas of Focus in Cloud Computing V2.1*, <http://www.cloudsecurityalliance.org/csaguide.pdf>
- [26] Giovanni Di Crescenzo, Niels Ferguson, Russell Impagliazzo, and Markus Jakobsson. *How to forget a secret*. In STACS, pages 500509, 1999.
- [27] Joel Reardon, Hubert Ritzdorf, David A. Basin, and Srdjan Capkun. *Secure data deletion from persistent media*. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, ACM CCS 13, pages 271284. ACM Press, November 2013.

- [28] J. L. Dautrich Jr and C. V. Ravishankar. *Compromising privacy in precise query protocols*. In Proceedings of the 16th International Conference on Extending Database Technology, pages 155166. ACM, 2013.
- [29] M. S. Islam, M. Kuzu, and M. Kantarcioglu. *Access pattern disclosure on searchable encryption: Ramification, attack and mitigation*. In NDSS 2012. The Internet Society, Feb. 2012.
- [30] X. Zhuang, T. Zhang, and S. Pande. *HIDE: an infrastructure for efficiently protecting information leakage on the address bus*. In ASPLOS 2004, pages 7284, 2004.
- [31] Daniel S. Roche, Adam J. Aviv, Seung Geol Choi. *Practical Oblivious Map Data Structure with Secure Deletion and History Independence*, IEEE Symposium on Security and Privacy (Oakland) 2016.
- [32] Hong Sun, Xueqin Zhang and Chunhua Gu, *Role-based Access Control Using Ontology in Cloud Storage*, International Journal of Grid Distribution Computing, 2014.
- [33] Zhen Mo, Qingjun Xiao, Yian Zhou, Shigang Chen, *On Deletion of Outsourced Data in Cloud Computing*.
- [34] Kevin D. Bowers, Ari Juels, Alina Oprea, *HAIL: A High-Availability and Integrity Layer for Cloud Storage*, Proceedings of the 16th ACM conference on Computer and communications security 2009.
- [35] Yang Tang ; Lee, P.P.C. ; Lui, J.C.S. ; Perlman, R. *FADE: Secure Overlay Cloud Storage with File, Assured Deletion, Dependable and Secure Computing*, IEEE Transactions Volume 9, 2012.
- [36] Christian Cachin, Kristiyan Haralambie, Hsu-Chun Hsiaoy, Alessandro Sorniotti,

Policy-based Secure Deletion, Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security.

- [37] Tim Mather, Subra Kumaraswamy, Shahed Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, O'Reilly Media, Inc. , 2009.
- [38] *Openstack Storage*, <https://www.openstack.org/software/openstack-storage/>
- [39] L Kissner, D Song , *Privacy-preserving set operations*, Advances in Cryptology-CRYPTO 2005 Springer.
- [40] D. Boneh, D. Mazières, and R. A. Popa. *Remote oblivious storage: Making oblivious RAM practical*. Technical report, CSAIL, MIT, 2011.
- [41] Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid *NIST Recommendation for Key Management Part 1: General (Revision 3)*, NIST Special Publication 800-57.
- [42] Qingji Zheng, Shouhuai Xu *Secure and Efficient Proof of Storage with Deduplication*, Proceedings of the second ACM conference on Data and Application Security and Privacy, 2012.
- [43] *Amazon Web Services*, <https://aws.amazon.com/>
- [44] Tim Mather, Subra Kumaraswamy, Shahed Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*, O'Reilly Media, Inc. , 2009.
- [45] L Kissner, D Song , *Privacy-preserving set operations*, Advances in Cryptology-CRYPTO 2005 Springer

- [46] O. Lassila, R. Swick and others, *Resource Description Framework (RDF) Model and Syntax Specification*, WWW Consortium, 1999.
- [47] *SWRL Rules*, <https://www.w3.org/Submission/SWRL/>

