Open access • Proceedings Article • DOI:10.1145/1555400.1555430

Using timed-release cryptography to mitigate the preservation risk of embargo periods

— **Source link** ↗

Rabia Haq, Michael L. Nelson

Institutions: Old Dominion University

Topics: Data security, Encryption, Metadata, Data loss and Cryptography

Related papers:

- Dynamic access-control policies on XML encrypted data

- Specifying Access Policies for Secure Content Dissemination of XML: A Technique Inspired by DNA Cryptography

- Secure Data Group Sharing and Dissemination with Attribute and Time Conditions in Public Cloud

- Secure Delta-Publishing of XML Content

- Poster: inference attacks against searchable encryption protocols

Share this paper:

View more about this paper here: https://typeset.io/papers/using-timed-release-cryptography-to-mitigate-the-2cmr4m05ko

Fall 2008

Using Timed-Release Cryptography to Mitigate Preservation Risk of Embargo Periods

Rabia Haq
Old Dominion University

Recommended Citation

USING TIMED-RELEASE CRYPTOGRAPHY TO MITIGATE PRESERVATION RISK OF EMBARGO PERIODS

by

Rabia Haq
B.S. August 2004, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
December 2008

Approved by:

_____

Michael L. Nelson (Director)

_____

Michele C. Weigle (Member)

_____

Ravi Mukkamala (Member)

ABSTRACT

USING TIMED-RELEASE CRYPTOGRAPHY TO MITIGATE PRESERVATION RISK OF EMBARGO PERIODS

Rabia Haq
Old Dominion University, 2008
Director: Dr. Michael L. Nelson

This research defines Time-Locked Embargo, a framework designed to mitigate the Preservation Risk Interval: the preservation risk associated with embargoed scholarly material. Due to temporary access restrictions, embargoed data cannot be distributed freely and thus preserved via data refreshing during the embargo time interval. A solution to mitigate the risk of data loss has been developed by suggesting a data dissemination framework that allows data refreshing of encrypted instances of embargoed content in an open, unrestricted scholarly community. This framework has been developed by exploiting implementations of existing technologies to "time-lock" data using Timed-Release Cryptology (TRC) so that it can be deployed as digital resources encoded in the MPEG-21 Digital Item Description Language (DIDL) complex object format to harvesters interested in harvesting a local copy of content by utilizing The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), a widely accepted interoperability standard for the exchange of metadata. The framework successfully demonstrates dynamic record identification, time-lock puzzle (TLP) encryption, encapsulation and dissemination as XML documents. This thesis dissertation presents the framework architecture and provides a quantitative analysis of an implementation. The framework demonstrates successful data harvest of time-locked embargoed data with minimum time overhead without compromising data security and integrity.

ACKNOWLEDGEMENTS

I would like to acknowledge Dr. Michael Nelson, my advisor, for his guidance, constant encouragement and infinite patience, without which this may not have been possible. He knew just when to nudge and when to push me through, and for that I am very grateful. I would also like to thank my Thesis committee, Dr. Michele Weigle and Dr. Ravi Mukkamala for their guidance and input.

I would like to thank my family for their unyielding support throughout my higher studies. Thanks to all my friends, especially Fatma, Shosho and OLH for their brainstorm sessions and helpful tips even when they did not understand a word of it. Their constant supply of midnight tea and encouraging words made this task seem easier.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

There is valuable scholarly research material that is difficult or impossible for researchers and practitioners worldwide to access. Lack of access to this research could be detrimental to the advancement of scientific research in various disciplines, or in improving the quality of health care. The traditional subscription-based journal model provides information at a cost, making it difficult to afford, especially by researchers in low-income countries. This access model has served as a catalyst for the Open Access (OA) movement, which is aimed at providing access to full-text journal articles online toll-free. Advocates of OA argue the subscription-based journal access system hinders free and open flow of ideas and information, while proponents of the traditional system argue that it governs and manages the flow of information, and ensures the maintenance of standard and structure of this information [66]. A balance needs to be achieved in order to ensure continuance of flow of scholarly information and coverage of publication costs by integrating the advantages contained in both these access models.

Embargoed access to academic material is a hybrid of the restricted, traditional access model and the toll-free, Open Access model. It is a temporary restriction imposed by the publisher on the full-text availability of the latest issues of a journal for a certain time-period, for example, two years, while the economic value of the journal is extracted. Individuals and institutions would have to subscribe to the journal in order to access the latest issues, while the previous issues are available to digital repositories and individuals without subscription cost. This access model has been formulated and adopted by various journals in order to cover their publication costs while supporting easy information access [77], a factor that was lacking in the OA cost-recovery model. This allows the publishers to generate revenue from their subscription business during the embargo period as well as include the journal articles in the aggregated databases to improve research accessibility to the scholarly community.

Various digital preservation methods are employed in Digital Libraries at present in order to ensure continuous existence, access and interpretation of data. No single

---

This thesis follows the journal style of *The Journal of Association for Computing Machinery*

digital preservation strategy is appropriate for all data types [45]. According to William Y. Arms,

"This is a time of prosperity in the United States; the next hundred years will surely see financial and political crises, wars, corruption, incompetence, and natural disasters. Tomorrow we could see the National Library of Medicine abolished by Congress, Elsevier dismantled by a corporate raider, the Royal Society declared bankrupt, or the University of Michigan Press destroyed by a meteor. All are highly unlikely, but over a long period of time unlikely events will happen." [2]

The fundamental idea of digital preservation is to not rely on a single method of digital preservation, and to utilize various strategies, such as data refreshing, migration [65] and emulation [68] to ensure data longevity and integrity. Data migration and emulation are out of the scope of this thesis research.

Data refreshing is the copying of bits to different systems that are distributed to various heterogeneous locations so that if one copy is destroyed by accidental or malicious means, other copies can be accessed to recover the local copy of the content. This method of digital preservation can be applied to preserve digital content that is freely accessible online, but data that has temporarily restricted, embargoed access cannot be preserved by this method.

Embargoed information is not accessible without cost before a predetermined time interval has passed. During this embargo period, users and researchers that have not subscribed to the embargoed journal are unable to access this information; data aggregators such as EBSCO Information Services[1] and Ovid[2], who assimilate content and further provide them to institutional libraries, are also unable to assimilate this content without subscription. Therefore, the refreshing method of digital preservation cannot be exploited as one of the digital preservation strategies to preserve this embargoed information, placing it at a risk during this embargo time interval, known as the "Preservation Risk Interval".

The purpose of this research is to address and mitigate this risk of data loss associated with embargoed information by suggesting a data dissemination model that allows data refreshing of embargoed content in an open, unrestricted scholarly community. It is also an effort to expand the flexibility of compromise between scholarly interest and commercial interest for those who are engaged, or wish to engage, in

---

[1]http://www.ebsco.com/
[2]http://www.ovid.com/

embargoed access. This model has been developed by exploiting implementations of existing technologies to time-lock and encrypt data using Timed-Release Cryptology [62] so that it can be deployed as digital resources encoded in the MPEG-21 Digital Item Description Language (DIDL) complex object format [79] to harvesters interested in harvesting a local copy of content by utilizing The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), a widely accepted interoperability standard for the exchange of metadata [35].

I.1 MOTIVATION

I.1.1 OPEN ARCHIVES INITIATIVE

Herbert Van de Sompel, then a researcher at the University of Ghent, was working in collaboration with researchers at Los Alamos National Laboratories, U.S., when he encountered the lack of a structural framework for the interoperability of distributed research journal articles. There was no single interface or protocol for searching across multiple repositories or for aggregating machine-comprehensible metadata for information sharing. In late 1999, a meeting was convened at Santa Fe, New Mexico, to address these issues. This convention resulted in the formation of the Universal Preprint Service, UPS, to initiate steps towards the identification and creation of some interoperable technologies to assist in dissemination of e-print archives [49, 12]. With the proliferation of e-print archives spanning various disciplinary subjects across the world, there was a need to effectively identify and harvest updated copies of research papers between repositories. Several workshops were held during 2000 to further explore the broadening scope of this communication problem and to explore an HTTP-based protocol for the exchange of metadata. The original mission of interoperability was improved and developed technically as well as organizationally, and was broadened to encompass the scholarly community, not just the e-print archive community. The Coalition for Networked Information and the Digital Library Federation funded the establishment of the Open Archives Initiative (OAI) that defined the framework named The Santa Fe Convention, which eventually evolved into the Open Archives Initiative Protocol for Metadata Harvesting, OAI-PMH [40].

I.1.2   OAI-PMH

The Open Archive Initiative "develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content" [78]. The OAI-PMH is a "low-barrier interoperability framework," [35] formulated to facilitate incremental harvesting of resources and related metadata and to enable resource discovery independent of the underlying architecture of the implementing digital repository. It has become the de facto standard for the exchange of metadata across the academic community [47].

I.1.3   RESOURCE HARVESTING

The advent of powerful and sophisticated search engines has enabled searching for data based not only on the metadata but also on the full text of research articles. Such developments in data discovery technologies require the harvesting of entire resource articles along with secondary information [70]. A protocol for exposing metadata as well as full text resources needed to be standardized in order to enhance searching capabilities and facilitate interoperability across distributed resources, bringing about the inception of resource harvesting within the OAI-PMH technical framework. This protocol functionality has been incorporated into an Apache server module, known as "mod_oai", to facilitate incremental web harvesting: crawling the world wide web to index or download content.

Presently, not only metadata, but other information pertaining to each record, such as usage history, is also harvested and disseminated to various repositories. This metadata does not contain explicit, efficient semantics for harvesting the resource itself from the provided information. Often the information provided on harvesting the resource from the provided metadata is not sufficient in resource location [35]. This makes it difficult for actual resource harvesting to be efficiently automated. A Complex Object Format for digital resource presentation has therefore been included in the OAI-PMH framework to allow efficient and accurate expression of secondary metadata, and to encompass the actual representation or reference of the digital resource itself.

The MPEG-21 Digital Item Declaration Language (DIDL), one of the complex object formats following the semantics of digital object representation described by the Kahn-Wilensky framework [32], has been incorporated with OAI-PMH in the

`mod_oai` module to accurately express metadata and represent the resource as a digital object.

I.1.4 TIMED RELEASE CRYPTOGRAPHY

This thesis research explores the concept of "Time-lock puzzles", first introduced by Timothy May in 1993, and later employed by MIT in 1999 for "The LCS35 Time Capsule Crypto-Puzzle" [64], to encrypt digital content in such a manner that it can only be accessed after a predetermined amount of time has elapsed.

Time-locked puzzles are encrypted pieces of information whose time to decrypt can be predicted since its decryption cannot be parallelized [42]. Therefore, in case the original resource is destroyed, the encrypted resource can be decrypted by performing serial computations on a dedicated machine that will take time "t" to decrypt the digital object, referred to as "timed-release cryptography."

I.1.5 ADDITIONAL MOTIVATION AND IMPLEMENTATION

The above-mentioned technologies can be exploited to mitigate the "Preservation Risk Interval" associated with embargoed digital objects. Digital resources can be

- time-locked for the embargoed time period,

- encompassed as complex digital objects, and

- efficiently exposed to service providers for data harvest.

In case of data loss, this harvested copy of encrypted data can be accessed after the embargo time required for breaking the encryption has elapsed. An updated time-locked digital object with weaker encryption can be disseminated at every harvest update, so that the time required to unlock and access these embargoed records decreases as expected in case of data loss.

Aside from the above-mentioned reasons that have motivated this thesis research, other factors that support this research are listed as follows:

- No known efforts thus far have been discovered that address the "Preservation Risk Interval".

- An analysis of the gravity of this problem and a viable solution are imperative towards finding a complete solution that addresses the need for preservation of present digital research media.

I.2   AIMS

The aim of this research is to provide a solution that would mitigate the preservation risk associated with embargoed resources, particularly in the research publication world. Embargoed access is a compromise between the economic and scholarly interest of the publication community, and this solution aims to enhance interoperability by facilitating a means of preserving the security and integrity of embargoed content.

I.3   METHODOLOGY

A complete "Time-Locked Embargo" framework has been formulated. Several evaluation tests have been performed to establish that the time required to break and decrypt these timed-release resources is linear, and cannot be reduced by parallel computation, hereby supporting this time-released encryption method.

Modifications in the OAI-PMH compliant Apache module, `mod_oai` have been made in order to incorporate the concept of dynamically time-locking embargoed resources before they are wrapped in XML files and disseminated to various repositories. Datestamps in these XML files have been modified in order to augment the searching and harvesting of these embargoed resources.

A modified version of the `mod_oai` Apache module has been installed and tested in order to evaluate this encryption and dissemination method. Harvest times of a website under embargo have also been gathered to ensure that time-locked data can disseminated with minimum time overhead.

I.4   THESIS ORGANIZATION

Chapter II provides a more detailed background of the technologies that are being exploited for this research project. It discusses the method by which the time-released algorithm can be used with datestamps in `mod_oai` to effectively provide incremental harvesting of embargoed resources to the harvester.

Chapter IV establishes the Preservation Risk Interval problem by introducing important nomenclature and further analyzes it by discussing the behavior of a repository containing embargoed content.

Chapter V explains the modifications and additions made to `mod_oai` in relation to the technologies discussed in Chapter II.

Chapter VI evaluates the performance of this research project based on the results of various tests performed during the course of the project. Chapter VII describes an optimization to the framework via "chunked" timed-release encryption.

Chapter VIII and IX offer recommendations for future improvements and concluding remarks to this project.

CHAPTER II

BACKGROUND

Digital Libraries are an evolving tool for the preservation and access of distributed digital media. Access to digital media is of utmost importance at the moment, as digital libraries and caches facilitate preservation and access to present media content. The concept of digital libraries and information retrieval pre-dates the development of the first computer. H. G. Wells was one was one of the first authors who visioned a central, format-independent "world brain" [81] that would enhance, or even replace, traditional libraries. Vannevar Bush's work [5] is considered a catalyst of change in the application of scientific research in the modern world. His innovative idea of automating human memory via associative linking resulted in the formulation of today's storage systems. Lesk [37] envisioned that all human knowledge would be available at ones fingertips by the year 2015. Further history and development of digital libraries can be found in textbooks [38], [3], [82] and in various other written material contributed by pioneers such as Stephen P. Harter [20].

II.1 PUBLISHING MODELS IN SCHOLARLY RESOURCES

A detailed examination of the three key types of journal access policies in the publishing community is required to better comprehend the various problems and issues prevalent in the journal community. Due to the constant evolution of journal publishing, it sometimes becomes difficult to differentiate and name the various access policies in use. This is clarified with the adoption of the use of Romeo Colors, where color codes are adopted to identify policy types and represent the taxonomy associated with the various forms of scholarly journal access [31]. The colors are:

- Red for traditional, subscription-based access,

- Yellow for embargoed access,

- Green for self-authored open access, and

- Gold for free and open access journals

II.1.1 EVALUATION OF SCHOLARLY PUBLISHING

Various problems have been identified in the publishing world that are hindering the efficient flow of published research.

According to Ulrichsweb[1], about 24,000 peer-reviewed journal articles are in print each year, publishing over 2.5 million articles, covering all disciplines and languages. Aside from the increase in journal unit prices by about 1% every year, libraries also have to cover their overhead costs [17]. For every $1 spent on acquiring journals, another $2 is spent on covering overhead [50]. Due to rising journal subscription costs and varied library budgets, it is impossible for any institution or library to be able to afford subscription to all these journals. Even if subscription to these journals was provided at cost price, it would be impossible to subscribe to all these journals. Therefore, libraries are able to provide only a fraction of these journals to its users, limiting the accessibility and thus the potential research impact of these journal articles.

According to Harnad et al. [17], the impact of a research article is the "degree to which its findings are read, used, applied, built-upon, and cited by users in their own further research and applications." Impact of a research article plays a great factor in determining the career, and thus the salaries, further funding and tenure of the authors. More impact leads to more citations of a research article, which also brings about prestige to the journal that published that research article. None of this would be possible without access and availability of this article, which makes accessing research journals an essential factor in all educational and research disciplines.

One of the solutions proposed to solve these problems is to make these full-text research articles available online for free, called Open Access, OA. One of the first Open Access journals of medicine, Journal of Medical Internet Research (JMIR) [2] emerged in 1998, with its first issue published in 1999. This established an electronic publication platform followed by PubMed Central[3] in 1999 and by BioMed Central[4] in 2000 for the spread of freely accessible postprint journal archives. In 2002, 13 original signatures at the Open Society Institute meeting resulted in the formulation of the Budapest Open Access Initiative (BOAI), a monumental achievement in the history of OA. The BOAI was the first collaborative initiative to define Open Access

---

[1]http://ulrichsweb.com/ulrichsweb/analysis
[2]http://www.jmir.org/
[3]http://www.pubmedcentral.nih.gov/
[4]http://www.biomedcentral.com/

as:

"free availability on the public internet, permitting any users to read, download, copy, distribute, print, search, or link to the full texts of these articles, crawl them for indexing, pass them as data to software, or use them for any other lawful purpose, without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself." [54]

As of April 2007, 4816 signatures have been added in support to the BOAI. This initiative has been supported by the release of Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities, and by the UN World Summit on the Information Society Declaration of Principles and Plan of Action in 2003 [55]. More recently, the National Open Access Policy for Developing Countries [51] was drafted in 2006, making OA an international collaborative amongst authors and publishers of developed as well as developing nations [72].

II.1.2   SUPPORT FOR OPEN ACCESS

Research is being conducted in order to estimate the significance of this problem. According to a study conducted by Lawrence [36] in the computer science discipline, the citation impact of conference articles that are freely accessible online in full text have a 336% higher impact than research articles that do not have "open access". The Université du Québec Ã Montréal, Southampton University and Universität Oldenburg are conducting an ongoing study comparing the access impact of OA accessible articles versus non-OA accessible articles across all disciplines over a 12-year sample of 14 million articles taken from the Institute for Science (ISI) database [18]. The study comprised of determining which of these articles, which were published in the same journal and in the same year, were OA and which were not available online without a cost. An analysis of the results for the physics discipline shows a greater positive correlation than the findings of Lawrence, with non-OA vs. OA journal citation ratios of 2.5 - 5.8.

One of the methods of providing open access to this material is by publishing in journals that provide immediate access to research publication online without any subscription costs, depicted with the color "gold". Another proposed method is by "self-archiving," depicted with color "green", where the author himself or herself makes their published article available on their personal website or in an institutional

repository. For example, Citeseer[5] harvests and aggregates articles from authors personal websites in the discipline of Computer Science, while arXiv[6] is a central disciplinary repository, established in 1991, consisting of articles in physics, mathematics, computer science and related disciplines. According to Harnad et al., self-archiving in the authors university institution archives has the greatest potential for promoting self-archiving. These archives should be OAI-compliant e-print Archives. In this way, these distributed archives can be made "interoperable," where users are able to search and browse through these archives and consequently retrieve the full text articles harvested in these repositories.

Further research regarding the access/impact can be conducted by counting the citations and the number of downloads of these articles. Performance predictors and indicators can be used to collect statistics for literature research from these repositories. Citebase[7], a download/citation correlator, is being utilized to observe article download frequency and its relative citations across a time period. The statistics collected are further employed to predict the citation of an article two years later. Such tools can be used by research funders and institutional evaluators for monitoring the progress of these articles and authors, for further decision-making and statistics, such as evaluating OA vs non-OA article access/impact and distribution of funds. According to a survey conducted by Swan and Brown in 2004 [75, 74], 39% of authors are already self-archiving their published material in one of the three methods available: either by self-archiving on their personal websites, in distributed university archives, or central disciplinary archives. During this research study, a vast majority of authors claimed that they would be willing to self-archive their published articles in a repository if their publisher or funding institution asked them to. Harnad et al. are promoting university distributed OAI-compliant archives for the purpose of promoting self-archiving, as it is the least costly OA model.

The demand and consequent acceptance of OA in the research community is evident in the inclination of research journals to go "green," which means that they officially allow their authors to self-archive their published articles. According to the latest Joint Information Systems Committee/Rights Metadata for Open archiving, JISC/RoMEO, survey of over 8,000 journals, over 90% of the journals are "green," while 5% of them are "gold," that is, they are Open Access journals, providing their

---

[5]http://citeseer.ost.psu.edu/cis
[6]http://arxiv.org/show_monthly_submissions
[7]http://citebase.eprints.org/analysis/correlation.php

content publicly toll-free.

Enhanced article impact affects the citation impact of the journal that published the OA articles as well, encouraging journals that wish to cooperate with the expanding OA movement. The percentage of OA journals rose from 55% to 83% from 2002 to 2003 alone and has risen to 92% by April 2007 [71]. More than 100 universities worldwide already maintain institutional e-print archives. Harnad asserts that implementing institutional Eprint archives for self-archiving would encourage the research community to adopt this model, for access availability as well as enhanced research impact of these published articles.

According to Harnad, the correct way of comparing the download and citation impact of OA vs. non-OA articles is to compare OA and non-OA articles in the same non-OA, "red" restricted journal [18]. Even though the journal itself does not conform to the Open Access model, it allows the authors to self-archive their articles. According to a study conducted by ISI, the citation impact of OA vs. non-OA articles is similar [57]. Out of the 8,700 journals indexed by ISI, 191 of them are OA with no discernible citation impact difference, equating them in comparison. This impact analysis reveals that OA journals do not produce low-quality articles, nor does providing OA lower the impact or the prestige of the journal itself.

According to research conducted by the NEC Research Institute, the number of citations of articles strongly correlates with its free accessibility and availability online [36]. An analysis of 119,924 conference articles in computer science and related disciplines indicates that the probability of an article being freely available online is a function of the number of citations of that article and its year of publication. More recent articles and highly cited articles have a higher probability of being available online, leading to the conclusion that making articles freely available increases the accessibility and consequently the citation and usage of that article.

The risks associated with the OA model have not yet been fully researched, and the long-term effect of this cost-recovery model has not been thoroughly explored, encouraging journals to go "green" rather than "gold" in response to the research communitys demand for OA. Authors in these "green" journals, such as those of the Journal of High Energy Physics, self-archive their articles in the journal, while publishers have agreed to self-archiving of these articles, leading to a collaborative effort towards promoting Open Access to the research community.

The advent of the OA model is bringing about an evolution in the traditional

cost-recovery model. Making information toll-free online makes it difficult for these "gold" journals to cover their expenses. These costs are sometimes covered by asking the author to cover the journal peer-review and publication costs of the published article.

An analysis of the citation impact of a research article is a complicated process involving various factors. Research studies support that Open Access results in greater citation impact, but other variables such as article duplication and authors' self-promotion on the Internet also need to be considered as factors leading to an increase in article availability, and thus an increase in the citation impact [11]. Providing Open Access would be insufficient in solving the access/impact problem by itself; an efficient infrastructure needs to be developed in order to train the researchers to effectively access and use the available information across various regions of the world. A balance needs to be achieved between these two access methods by experimentation and a systematic evolution in the exchange of research information.

II.1.3 EMBARGOED ACCESS: HYBRID APPROACH

Evolving attitudes in access of information are paving the way for new data access models. Publishers and societies are striving to optimize the free-flow of information in the academic community, while covering the costs of publication. Embargoed access, depicted with the Romeo color "yellow," is an emerging publishing model that provides readership and access to the research material at a cost, for a certain amount of pre-determined time, while the publishing costs are extracted. After the pre-determined amount of embargo time period has elapsed, the scholarly material is available toll-free. This access method provides an alternative to the cost model of Open Access, while advocating the flow of required knowledge for advancement in science and humanities, making access to this pool of information freely available via the web after the embargo period has passed.

This time-delayed access to the latest issues of an embargoed journal actually increases the total number of full text articles that become accessible to scholars. This is because of the influx of now freely available, peer-reviewed previous issues of these journals, which become part of repositories. These embargoed articles can still be indexed and abstracted, and consequently found by search engines during research by purchasing the required articles or by following the citation links to an electronic journal subscription. Electronic availability of journals makes the academic

community aware of its usefulness and existence, and contributes to interoperability between repositories.

Online accessibility should not be substituted for print subscription to a journal. Electronic availability of a journal enhances the research and access capabilities of the research because of various efficient searching facilities, and accessibility of an article from citation links. Paper subscriptions should be based on the usage of the journal and its prestige, thus keeping the cost revenue from subscriptions secure for the publishers.

II.2 DEFINITION OF THE "PRESERVATION RISK INTERVAL"

The research dissemination effects and economic feasibility of the open access publishing model are still being explored. A long term analysis of the impact of open access scholarly material is required to further establish the viability of this business model. Embargoed access is being adopted by various research journals and publishers for the purpose of providing the research community with free content after the economic value of the research is extracted through the traditional subscription system during an embargo period. PubMed Central (PMC) digital archive[8] currently contains 497 journal titles, out of which 24% are embargoed titles with embargo period ranging from one month to thirty-six months. *The New England Journal of Medicine* published by Massachusetts Medical Society has a 2007 impact factor of 52.589 with an imposed embargo of six months. *Genes and Development* published by Cold Spring Harbor Lab Press, Publications Department, has an ISI impact factor of 14.795 in 2007 with an embargo period of six months. *The EMBO Journal* by Nature Publishing Group, has an impact factor of 8.662 in 2007 with twelve months of embargo. A comprehensive list of embargoed titles could not been found, but readers can search for "embargoed journal titles" in Google to discover journal lists of various digital archives and databases.

Since full-text embargoed journal articles cannot be harvested by distributed repositories during this embargo period, copies of this research material are not disseminated to other digital libraries that may later be accessed in case the local copy of the article is irretrievable. This places a digital preservation risk on research material that is under embargo.

---

[8]http://www.pubmedcentral.nih.gov/fprender.fcgi?cmd=full_view

The "Preservation Risk Interval" is the time period when the scholarly research material has been published but is under embargo for a predetermined amount of time. During this time frame, metadata pertaining to the published article is available to the entire community for indexing purposes, but the full-text article can only be accessed by subscribed users and repositories, leading to embargoed data diffusion only within a subset of the research community. In case the original copy of the research material in the local repository is destroyed by natural or malicious causes, a reliable copy of the destroyed data may not be available in the subscription-based research community. Under such circumstances, another copy located in a distributed, heterogeneous repository is required to refresh the destroyed local copy. Thus, limiting the distribution of embargoed content to within a subset of the research community is hindering digital preservation by data refreshing. Data refreshing by means of frequent backups of original data is advisable, but is insufficient for the purpose of long-term digital preservation, because faults in the original data system may not be independent of its replicas [4]. Therefore, the digital preservation method of data refreshing via content dissemination to various distributed repositories needs to be exploited as a digital preservation method for embargoed records along with open access records. In this manner, the destroyed original copy of the data in embargo may be refreshed by another repositorys copy of the record in case a reliable copy within the subscription-based community cannot be retrieved.

Embargoed data, along with open access research content, can be provided for harvest using the OAI-PMH protocol via `mod_oai`. The integrity of the embargoed content can be maintained by ensuring that the embargoed data is not accessible until the specified embargo time period has elapsed. This can be achieved by time-locking the embargoed record using Timed Release Cryptography.

II.3   TIMED-RELEASE CRYPTO - THE DATA MODEL

In 1978, Rivest, Shamir and Adleman (RSA) proposed to reconstruct two large prime numbers from their product, considered to be an NP-hard problem, as a data cryptology system [61, 30]. The RSA cryptosystem is a widely-accepted and patented public-key scheme that is commonly used to exchange keys, create digital signatures and encrypt messages [76].

Conventional key-generation algorithms, such as the RSA encryption algorithm,

have encryption keys that are a composite of two large primary numbers of approximately equal size such that,

$$n = pq \tag{1}$$

where this modulus n is used to produce both the public and private keys. It consists of modular arithmetic used to calculate the factors of large numbers whose complexity is dependent on the size of the numbers used. The complexity of the puzzle generated is just the complexity of the factoring problem. If brute force were to be applied to break the key, estimated computation time can be reduced by k, by dedicating k computers to compute the product in parallel, thus reducing the time required to access the encrypted resource.

For timed release crypto, a non-parallelizable key encryption system is proposed, where

$$t = TS \tag{2}$$

is calculated, T being the amount of time in seconds for which the puzzle is to be time-locked. A random key can be generated using a conventional cryptosystem, where the key may consist of enough bits to be considered sufficiently complicated, and cannot be easily examined and broken. Calculating inputs n using equation 1, and a random a, where $1 < a < n$, the encryption key can be encrypted as

$$Key\ encryption = Key + a^{2^t}(mod\ n) \tag{3}$$

The output of the time-lock puzzle is thus (n, a, t, encrypted key, encrypted resource). Input variables, such as p and q that were used during the computation of the puzzle, are destroyed.

Without p and q, searching for the key itself to solve a puzzle that is encrypted using this method is impractical, rendering brute force methods of breaking the encryption key non-parallelizable. The most efficient method of retrieving the encryption key would be to calculate the variable

$$b = a^{2^t}(mod\ n) \tag{4}$$

initially used during key encryption. This can be achieved by sequentially performing t squarings on the value a. This would require a dedicated computer performing continuous, sequential computation for approximately t time units to decrypt the key and thus break the encryption. This encryption method can be applied with

various, carefully selected values of time unit t to encapsulate the resource for a pre-determined amount of time, which is equivalent to resource embargo period, ensuring that the resource is not released until the desired computation time t has elapsed.

II.3.1 TIMED-RELEASE CRYPTOLOGY - AN ALTERNATIVE AP-PROACH

Another existing approach to Timed-Release Encryption (TRE) is the use of a third party intervention, or Trusted Agents (TA). The use of TAs to utilize timed-release cryptology to encapsulate sensitive information was also first approached by Timothy May, and later elaborated by Rivest, Shamir and Adleman as an alternative approach that does not require usage of the receiver's computation resources. A time-server(s) intercedes as the negotiator between the content source and the client who is trying to access the content. During content encryption, a pair of private keys is produced. The client receives an encrypted instance of the content, as well as one of the private keys, later used for identity verification. The other key from the pair, required for successful decryption and accessibility of content, is received from the time-server only after the required embargo period has elapsed [16, 8]. Various protocols have been developed for successful and secure communication between the time-server and the user. Initial protocols required a lot of TA-user interaction, which compromised the anonymity of the TA. The Conditional Oblivious Transfer Protocol [9] was then proposed that provided anonymity to the content provider, but not the client during communication. Further approaches, such as New-TRE [23] propose non-authenticated, non-interactive server-based schema that may be implemented using multiple time-servers for further anonymity. This TA-based scheme is being manipulated for public-key encryption and non-interactive and secure server-client interaction across the Internet.

Timed-release encryption using time-lock puzzles (TLP) has been implemented in this research because it does not require any outside intervention or dependency for authorization or further information interaction during decryption time. Suppose the embargo period for a puzzle is significantly large, such as 35 years for the MIT/LCS time-lock puzzle [64], which would require significant overhead to ensure the security and continual access of trusted agents till the embargo period expires. Recent trends in the use of information and resources demand the creation of independent, self-contained digital objects for long-term preservation. Successful preservation of locked

content requires implementation methods that are able to survive large embargo periods. This thesis research explores the use of time-lock puzzles for preserving data integrity and its application in digital libraries as an add-on to `mod_oai` for the purpose of digital preservation of embargoed data.

II.4 OAI-PMH - THE REFERENCE MODEL

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) facilitates interoperability and extensibility between data and service providers. It was formulated to address the need of a protocol that would contain update semantics, so that a harvester would be able to identify and request data from a repository from the last date of harvest. A six-verb protocol has been incorporated into the `mod_oai` apache module to address this need of selective metadata harvesting through OAI-PMH.

II.4.1 "REGULAR OAI-PMH"

OAI-PMH is a simple HTTP-based, request-response transaction protocol between a data provider (a repository), and a service provider (a harvester), for the exchange of content. This technical framework of OAI has been adopted by digital repositories to expose metadata about their resources, which can be extracted by harvesters using this protocol. It allows multiple-disciplinary as well as update-centric resource discovery, while facilitating repository synchronization and federated search [73].

Upon a harvester request to a repository, the data returned to the harvester is related to the local collection of the server-end repository. The repository responds with an appropriate XML document consisting of information regarding its resources to the harvesters request. These resources are represented in XML through OAI-PMH defined entities, such as a *record, header*, *metadata, item* and *set*. A *record* entity contains all information pertinent to one resource. *Header* encompasses other related information of the resource, such as unique *identifier*, *set* and *datestamp* associated with the metadata of the resource. An *item* contains all metadata related to one resource. A *record* contains metadata about one resource. Thus, an *item* can contain more than one record, each containing metadata in a specific metadata format pertaining to one resource. This hierarchy is further clarified by Figure 1 taken from [46]. *Sets* are repository-defined collections of these *items*. Large XML
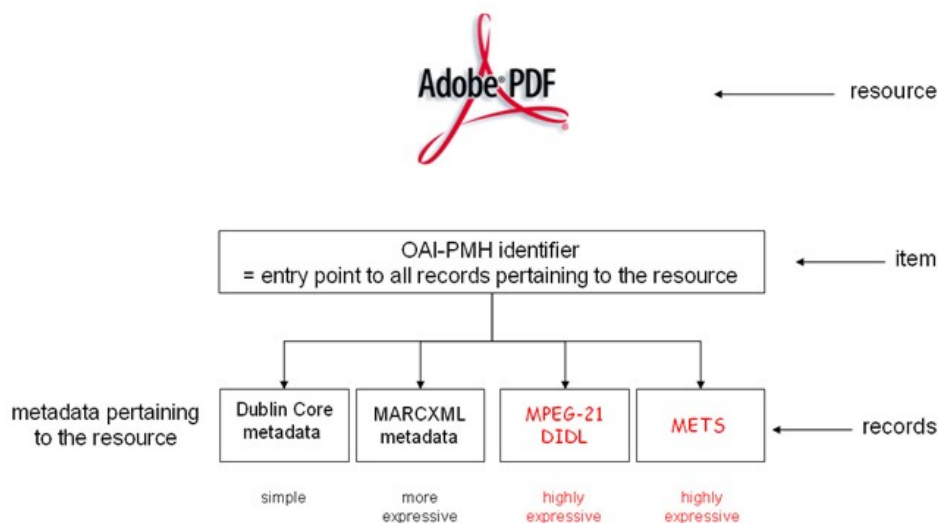
FIG. 1: OAI-PMH Data Model.

documents are segmented into multiple HTTP transactions with a *resumption token* used in a secondary request to receive the next part of the pending results.

The protocol consists of six protocol requests, commonly referred to as "verbs," that allow interaction between the repository and the harvester, which are listed in table I, taken from [46].

Verb	Comment
Identify	returns a description of the repository (name, POC, etc.)
ListSets	returns a list of sets in use by the repository
ListMetadataFormats	returns a list of metadata formats used by the repository
ListIdentifiers	returns a list of identifiers (possible matching some criteria)
GetRecord	given an identifier, returns that record
ListRecords	returns a list of records (possibly matching some criteria)

TABLE I: 6 OAI-PMH Verbs.

Three auxiliary verbs in the protocol have been introduced to determine the nature of the repository. *Identify* provides the harvester with preliminary information such as granularity, administration, etc. *ListMetadataFormats* lists the available metadata formats and *ListSets* retrieves the disciplinary-based structure of the repository. Three further protocols facilitate actual data transaction between the

harvester and the repository. *ListRecords* exposes information of the *records* contained in the repository. *GetRecord* retrieves a single *record* from the repository, and *ListIdentifiers* lists all *identifiers* of the *records* contained.

Multiple metadata schemes are supported in OAI-PMH. Metadata representation in Unqualified Dublin Core (DC), the lowest denominator, is mandatory by all data providers in order to enforce interoperability amongst distributed repositories, but they are free to support other, more expressive XML-transportable metadata formats. The protocol provides data providers with the extensibility and flexibility to mold their applications according to the requirements and needs of the data being supported. The harvester can also request metadata in a specific metadata format, that may support complex digital objects, by using metadata prefixes "http" or "didl" in `mod_oai`.

II.4.2 RESOURCE HARVESTING WITHIN THE OAI-PMH USING MPEG-21 DIDL

The Apache `mod_oai` module has been designed to implement the OAI-PMH functionality in a network-based environment. It provides selective harvesting through *from-until* parameters, so that only the desired metadata may be harvested from a data provider. This feature has enhanced efficiency by facilitating harvest of data from the previous repository update instead of a complete repository harvest with redundant copies of records that may already exist in the harvesters data collection [78, 46]. This is achieved by set-based harvesting or by date-based harvesting of records. Set-based harvesting is achieved by specifying the *set* parameter in the harvest request. Date-based harvesting is achievable through the *from-until* date parameters in the OAI-PMH request, whereby only those records whose *last-modified* dates lie between the provided parameters are included in the response.

This framework also allows flexibility in representation of resources, such as extension to inclusion of the resources themselves, accompanied with harvest of metadata about the resources. Often, service providers prefer to download the resource itself, along with the metadata associated with digital resources. This is to have a local copy of the record, in case the original instance is no longer accessible, and to provide more comprehensive searches, such as in-text searching. OAI-PMH facilitates this process using the MPEG-21 DIDL complex object format to accurately represent resources assimilated within XML documents to be disseminated to service providers.

Various standards for the XML-based representation of resources, termed Digital Items, exist in the scientific community. The OAI-PMH framework has adopted the Moving Picture Experts Group-21 Digital Item Declaration (MPEG-21 DID) ISO standard [58] abstract data model for accurate representation of resources. The MPEG-21 Digital Item Declaration Language (MPEG-21 DIDL), an XML instantiation of this data model based on the DID entities, has been derived to express these complex digital objects. A Digital Item that is described using this defined data model and represented in the DIDL syntax is referred to as a DIDL document.

The MPEG-21 Abstract Model, although complex and verbose, provides the flexibility to accurately describe and encompass a digital object and related metadata. This is achieved via representation of these digital objects through various entities, each entity instantiated as an XML element in the DIDL XML Schema [28]. The *resource* entity is the actual, identifiable datastream, either digital or non-digital, such as a picture, a video clip, a text document, or a painting in a museum. A *component* entity is used to group together *resources* and related information about these *resources* wrapped in a *descriptor/statement* entity construct. A DID *item* is the point of entry for information pertaining to one *resource*. It groups together one or more *descriptor/statement* constructs. *Items* may contain other *items*, each representing one instance of the *resource* [29, 21]. Figure 2 is a graphical view of a resource expressed in MPEG-21 DIDL.

The MPEG-21 DIDL complex object format provides the syntax for resources to be accurately expressed and reliably deployed through `mod_oai` during incremental resource harvesting. The framework allows sufficient flexibility for the DIDL syntax to be modified to include further information related to the resources, such as various metadata formats or copyright information. This flexibility is exploited in this research to affectively express time-locked records and pertinent information to decrypt and access these time-locked records, as later described in Chapter V.

II.5 SUMMARY

This chapter introduced the existing technologies that build the framework of this thesis research. Timed-Release Cryptology has been utilized to encrypt embargoed content during data dissemination in `mod_oai`, and the MPEG-21 DIDL complex object format incorporated in the Apache module has been modified to accurately encapsulate and express embargoed, time-locked digital resources. The next chapter
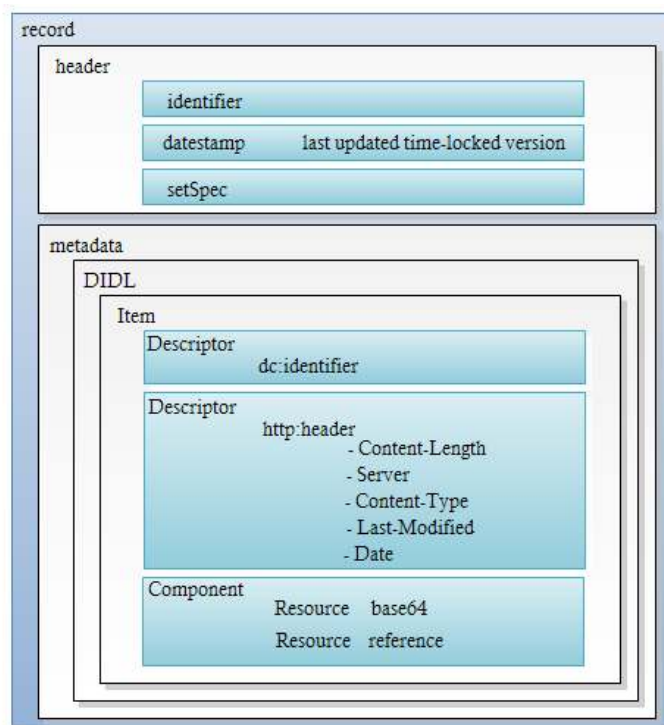
FIG. 2: Structure of a resource expressed in MPEG-21 DIDL Abstract Model.

analyses how similar solutions currently address the "Preservation Risk Interval" associated with embargoed content and establishes a need for a more robust and efficient solution to this problem.

CHAPTER III

RELATED WORK

Various digital preservation systems and tools have been deployed to ensure continual access to published content. With the plethora of electronic journal publishing replacing or augmenting traditional paper journals, it is imperative that steps be taken to ensure preservation of, and persistent access to, this published content. This chapter assesses a variety of digital preservation solutions and analyses whether they address the digital preservation of embargoed scholarly content.

III.1 LOCKSS

III.1.1 AN OVERVIEW

Lots Of Copies Keeps Stuff Safe (LOCKSS) is a persistent web-publication preservation system that incorporates the "purchase and own" library methodology in loosely coupled distributed digital repositories [59]. LOCKSS is a tool designed to provide support to research libraries for the purpose of creating and maintaining caches of subscribed electronic content to facilitate preservation and continual access of this material in case the content is no longer accessible from the publishers repository.

III.1.2 DIGITAL PRESERVATION

LOCKSS has tried to conform to the needs of the library as well as the user and publisher community. It allows libraries to establish local, low cost and maintenance 'caches' that archive a local copy of the journal articles residing on the subscribed publishers repository. In case the reader is unable to access the content through the publishers archive, the content can be retrieved from the local cache copy to provide the reader access to the research material.

The system complies with the needs of the publishers by augmenting the collection of reader usage and interaction data. The cache data is available only as a fallback system in case the publishers repository access fails. The cache content is subjected to the same legal agreement as the original journal subscription to make sure that readers are not distributed any data that they do not have access rights to, thus maintaining the integrity of the content [69].

LOCKSS proposes a highly replicated peer-to-peer system, whereby digital preservation of material is secured through multiple copies of the same content in geographically disparate repositories. This network of repositories participates in "rate-limited sampled voting" [41] to automatically detect and repair a defected copy of the document by replacing the damaged copy with an undamaged instance of the document located at another LOCKSS cache. Only that document which was previously present in a LOCKSS cache is replaced, so that no document is illegally replicated or leaked to unauthorized users. The system frequently crawls the journal publisher's website and harvests any new subscription material, storing a local copy of the data in the cache. This harvested content can further be included in the institution's provided indexing and searching facilities [67, 60].

III.1.3   PRESERVATION OF EMBARGOED CONTENT

The LOCKSS system accomplishes the goal of preserving peer-reviewed journal articles of the implementing institution's interest. This data is distributed amongst other LOCKSS systems and readers that have already purchased access rights to paid access, traditional journals or embargoed journals. This enforces data integrity and preservation via data refreshing and replication, but the system operates only amongst subscribed readers, a subset of the electronic journal usage community.

Although this system is successful in accomplishing its purpose of providing subscribed readers with persistent data access and preserving digital bits, it is limited to a subset of the scholarly community. It diffuses digital content only to readers that have already purchased the right to embargoed content. This not only confines the efforts of preservation via data refreshing to the subscription-based community, but also makes the embargoed content vulnerable, since the number of copies of the same journal article is dependent on its popularity and the number of subscriptions. There is a need for a more robust and reliable system that can deploy an encrypted copy of the journal article to the scholarly community, achieving far reaching preservation effects.

III.2   PUBLISHER-DRIVEN PRESERVATION INITIATIVES

Two preservation techniques are known in the publishing community that involve publishers as active role players in digital preservation efforts of scholarly material.

III.2.1 CLOCKSS

Controlled LOCKSS[1] is the implementation of the LOCKSS initiative within the academic community where publishers have collaborated with research libraries located in heterogeneous locations for the preservation of digital academic publishing. This is a collaborative preservation effort between 11 publishers and 10 renowned library institutions worldwide to maintain distributed dark archives to provide continual access to research publication, and is going live in January of 2009. This maintained dark archive can be triggered upon trigger events, such as

- cancellation of an e-journal title

- e-journal no longer available from a publisher

- publisher ceased operation

- catastrophic hardware or network failure

Upon occurrence of these events, the CLOCKSS archive can be utilized to ensure and provide persistent access to research material that be otherwise be inaccessible from the publisher or other institutions. The organization then votes to determine whether the triggered content should be made freely available to the academic community.

III.2.2 PORTICO

Portico[2] was launched as a preservation initiative in 2005 as a collaboration between JSTOR and Ithaka and is supported by the Andrew W. Mellon Foundation and The Library of Congress. It is a cooperative between 66 publishers and 477 international library institutions that maintain dark archives of "raw" digital content to provide permanent access to scholarly journal material. Portico accepts source files of published content, such as XML, SGML or PDF documents of the journal article and converts it to a normalized format aimed at providing long-term access rather than immediate access [13]. Therefore, in case of a triggered event, Portico provides delayed access to retrieved content.

---

[1] http:www.clockss.org
[2] http:www.portico.org

III.2.3   DIGITAL PRESERVATION

As of 2008, two trigger events have been recorded where triggered content hosted by CLOCKSS and Portico has been retrieved and voted by the organizations to become available online for free.

The journal *Auto/Biography* ceased to publish in 2006 under SAGE Publications. This journal title was hosted by IngentaConnect until 2008, after which it ceased publication. This triggered content has now been made available toll-free in the CLOCKSS and Portico repositories to provide continual access to this web-based academic content. Similarly, access to *Graft*, that ceased publication in 2003, is also provided by CLOCKSS and Portico.

The dark archives maintained by CLOCKSS and Portico are successful in providing continual access to material that has ceased publication and may be inaccessible via the publisher or any other source. Both preservation techniques maintain a limited number of reliable, independent and heterogeneous archives that differ in technological and governance models [45]. Portico has secured higher publisher backing and library institution participation, but provides a costly preservation solution. CLOCKSS implements the LOCKSS program with a less economic overhead and provides perpetual access to content but is only in the initial stages of development. Sustainability of the archiving solution is imperative in providing perpetual access and long-term digital preservation of content. Therefore, various digital preservation techniques need to be employed for preservation of scholarly digital media.

III.2.4   PRESERVATION OF EMBARGOED CONTENT

Both CLOCKSS and Portico preservation initiatives provide publisher-driven preservation solutions of academic material by maintenance of dark archives that are triggered only upon occurrence of a trigger event. These preservation techniques are limited to the participating institutions and publishers, and therefore provide a preservation solution only for a subset of the digital media. An enhanced cooperative infrastructure between publishers and institutions is required that establishes concrete rules for the type and volume of preserved content and the preservation techniques employed to provide perpetual access as well as long-term preservation of electronic content.

III.3 RISK ASSESSMENT AND DECISION SUPPORT SYSTEMS

III.3.1 AN OVERVIEW

Multiple solutions are required to solve the problem of preservation of diverse, hetero-geneous collections of digital content. Various services that integrate these numerous preservation methods and tools into one interface are being developed to provide decision support and recommendation services for repositories.

Various efforts are underway to mitigate the risk of data loss associated with digital content.The Virtual Remote Control (VRC)[3] system has been developed by Cornell University as a risk discovery tool for identifying potential preservation risk associated with digital collections [43]. Creative Archiving at Michigan & Leeds: Emulating the Old on the New (CAMiLEON)[4] is a collaborative effort between the Universities of Michigan and Leeds, UK, for exploration of emulation as a digital preservation method. Preserving and Accessing Networked Documentary Resources of Australia (PANDORA)[5] and the PANDORA Digital Archiving System (PAN-DAS)[6] are initiatives of the National Library of Australia in efforts of data encapsu-lation, management and long-term access to one of the countrys most comprehensive digital collections [56, 34].

IBM's strategy for ensuring long-term access to digital content is the visualization and implementation of a platform-independent Universal Virtual Computer (UVC) approach. This project utilizes migration and emulation preservation methods to encapsulate the digital data in a Logical Data Schema (LDS) with related metadata to create a hardware, software and format independent object that can be constructed and accessed in the future using the UVC emulator and viewer [22, 48]. The proof of concept for this system has been implemented at Koninklijke Bibliotheek (KB), the National Library of the Netherlands [39]. The digital archiving system of KB [53, 80], known as e-Depot, has been deployed in efforts of a digital archiving solution for electronic publications. It has been developed in compliance with the OAIS reference model [1] and utilizes the IBM system known as the Digital Information Archiving System (DIAS). It contains a Preservation Manager component within the infrastructure that semi-automates the digital object rendition, registration of

---

[3]http://prism.library.cornell.edu/VRC/
[4]http://www.si.umich.edu/CAMILEON/about/aboutcam.html
[5]http://pandora.nla.gov.au/
[6]http://pandora.nla.gov.au/pandas.html

metadata and development of tools for accessibility [52].

The DIgital Library Infrastructure on Grid ENabled Technology (DILIGENT) project, partially funded by the European Commission, is an amalgam of Digital Libraries and Grid Technology. The DILIGENT infrastructure enables sharing of resources between various user communities simultaneously. The grid framework provides the platform for implementation of a network of diverse, virtual, on-demand digital libraries, and allows implementation of related functionality, such as searching, data rendering, annotation and personalization. It facilitates interoperability between repositories and allows sharing of content and computing resources, enabling implementation of applications that may otherwise have resource limitations [7, 6].

Conversion and Recommendation of Digital Object Formats (CRiB)[7] is a three-component system that has combined a Service-Oriented Architecture (SOA) with Web Services to provide a platform for data encapsulation and solution recommendation services [15]. The Migration Broker provides an interface to various migration services to access the preservation risk of a digital object through established evaluation criteria. The Format Evaluator produces relevant information regarding an object format's stability and prevalence by comparing it to a Format Knowledge Base. This information can aid in the recommendation process by determining which available format would provide the highest preservation gain. A suitable format can then be selected for migration of that digital object. The third component, the Object Evaluator, evaluates the original object format with the outcome of the converted format to calculate the success rate and quality of performed migration service [14]. This system can be utilized to enhance the decision process during migration to achieve long-term accessibility of digital objects.

A similar but more versatile and flexible approach has been adopted by the Preservation webservices Architecture for Newmedia, Interactive Collections and Scientific Data (PANIC)[8] system, which is not limited to the digital collection or the type of preservation method. PANIC exposes various tools, systems and services as Semantic Web services to provide a three-step preservation system to facilitate preservation risk detection, notification and recommendation services of mixed-media objects, and their invocation with minimal human intervention [27, 24]. These three steps are:

- Preservation Metadata Capture

---

[7]http://crib.dsi.uminho.pt/
[8]http://www.itee.uq.edu.au/ eresearch/projects/panic/index.html

Consists of tools and services designed to generate and capture descriptive metadata for the digital object.

- Obsolescence Detection and Notification

  Contains format and software registries that provide information regarding the most recent and widely used standards and formats. They can be used to compare the metadata of each digital object to detect preservation risk, or format obsolescence. A notification of the results can be sent for further recommendation and calculated action.

- Preservation Discovery and Invocation

  The recommendations made by the Detection services can be implemented using this component of the system. A manager can specify the attributes of the required preservation service on an object. Using Semantic Web services, a Discovery Agent can then be dynamically deployed to find migration or emulation preservation strategies from the pool of available services by comparing the attributes specified [26, 25]. Further human intervention is required to select and invoke the most appropriate preservation service from a list of recommendations collected. Provenance metadata for the updated object is also expanded to include the changes implemented.

III.3.2 DIGITAL PRESERVATION

The systems described above are easily expandable and can be used to include a variety of preservation techniques. They provide a platform that allows the convergence of existing preservation techniques to facilitate the implementation of the preservation process in repositories. They can be utilized as tools that would assist in preservation efforts by providing unified access to a plethora of techniques, but do not themselves provide a conclusive preservation service to solve the problem addressed in this thesis research. The metadata encapsulation services provided by these systems would assist in efficient data management and exchange within and between repositories, but it does not expand the scope of replication to outside the selected community.

III.3.3 PRESERVATION OF EMBARGOED CONTENT

The proposed solution to the preservation of embargoed content suggested and implemented in this thesis research can be incorporated as a preservation technique in this class of systems to impact the scope of data sharing between institutions implementing the described systems. These systems integrate existing preservation technologies into one service to provide a collaborative solution to the problem of digital preservation of complex digial objects and do not address the preservation of embargoed content itself.

III.4 SUMMARY

This chapter scrutinized some of the existing preservation systems and tools that are currently employed in the digital community to provide preservation solutions. It was concluded that the discussed systems were focused towards providing long-term access and preservation solutions for a subset of the digital community and do not address preservation of embargoed content.

CHAPTER IV

THE "PRESERVATION RISK INTERVAL"

IV.1   THE "PRESERVATION RISK INTERVAL" PROBLEM

IV.1.1   INTRODUCTION

The perception towards digital preservation is under constant evolution. Emerging preservation systems are drifting towards open architectures where various distributed systems may be networked together to trigger notifications and preservation techniques, such as replication, migration and emulation. This innovative approach signifies that preservation of scholarly material via replication is no longer limited to supporting institutions, but is diffusing to a much broader and generic community of interest. This requires a more robust content dissemination system that can be expanded to include dissemination of embargoed content.

IV.1.2   BEHAVIOR OF A REPOSITORY

Consider a repository that contains numerous records with varied embargoed time periods. This could be a case in a large repository that contains journal articles from different journal issues where the latest issue contains embargoed content, while the previous issues of the journal are open access articles. As a result, the records contained in the latest issue need to be identified and time-locked before data dissemination. The embargo period of the records would begin at the time of article publication. The repository provides a predetermined number of instance updates of each record with successively weaker time-locks. Thus, every record update is a weaker time-locked encryption that requires less time to break and consequently access the data. The last instance update would be an unlocked version of the record at the end of the embargo time period. This scenario is further illustrated in Figure 3.

IV.1.3   ANALYSIS AND MITIGATION OF THE "PRESERVATION RISK INTERVAL"

Time-locking embargoed records in an institutional repository for the purpose of digital preservation during data dissemination introduces an overhead. This is the

time required to unlock the records in order to effectively access a decrypted version of the records. The establishment of nomenclature, as described in Table II, is thus imperative for assessing the amount of time required to effectively unlock and access the time-locked records held in the repository.

i	time unit
$R(i)$	number of records in repository at time i
R_{update}	number of record updates to repository per time unit
$R(0)$	initial number of records in repository at time $i = 0$
$H(i)$	number of records at harvester at time i
H_{update}	harvester update frequency per time unit. $H_{update} = 0.5$ means data harvest every 2 units
$H(0)$	initial number of records at harvester at time $i = 0$
$publisher_{start}$	publisher-imposed global datestamp after which all records published in the repository are under embargo
$embargo_{start}$	start date for record embargo period, which is record creation date
$embargo_{end}$	end date for record embargo period, after time period $embargo_{length}$ has elapsed
$embargo_{length}$	embargo time period for each record
$embargo_{decrement}$	number of times the record time-lock is decreased. This is also the total number of record updates
r_x	record with embargo period = x, with x = 0 as an unlocked record

TABLE II: Variable definitions.

Let us assume the existence of a repository R that consists of a set of records such as $R = \{a_w, b_x...k_z\}$, where the subscript for each record instance is the remaining embargo time period for that record. This repository R contains an initial number of $R(0)$ records at time $i = 0$. The records are initially time-locked for a total embargo period $embargo_{length}$, with a predetermined number of $embargo_{decrement}$ embargoed record updates. The embargo period begins at the time of record publication, which is at datestamp $embargo_{start}$. This repository is updated at every time unit i when the existing records in the repository are updated with a new embargo period time-lock. All records in the repository will be updated with a new embargo period with every repository update. The repository also grows linearly, by R_{update} records with every update.

An example and its analysis are elemental in better comprehending the contribution of these variables for calculating the required computation time. Consider the scenario where repository R has

- $R(0) = 5$

- $R_{update} = 1$ new record is published with every repository update.

- total embargo period for each record $embargo_{length} = 3$ months, and

- $embargo_{decrement} = 3$ is the number of times the record time-lock is decreased, before an unlocked instance is published.

An updated instance of each record is published every time unit the repository is updated. Table III describes the number of records and their embargo period with respect to time.

a_3, b_3, c_3, d_3, e_3	$i = 0$	time $i = 0$ with $R(0) = 5$ records
$a_2, b_2, c_2, d_2, e_2, f_3$	$i = 1$	5 records updated, 1 new record published
$a_1, b_1, c_1, d_1, e_1, f_2, g_3$	$i = 2$	6 records updated, 1 new record published
$a_0, b_0, c_0, d_0, e_0, f_1, g_2, h_3$	$i = 3$	unlocked records at $i = embargo_{length} = 3$
$\vdots$		
$a_0, b_0, c_0 \ldots p_0, q_0, r_1, s_2, t_3$	$i = 15$	$R_{update} \times embargo_{length} = 3$ records locked any time when $i > embargo_{length}$

TABLE III: Behavior of an active repository with embargoed records, with $R(0)=5$ records, $embargo_{length}=3$ months and $R_{update}=1$ record.

Therefore, the total number of records held in the repository at time $i = 15$ is equal to

$$R(i) = R(0) + i \times R_{update} \tag{5}$$

$$= 5 + 1 \times 15 \tag{6}$$

$$= 20 \tag{7}$$

As the repository begins with an initial number of records $R(0)$, which have been time-locked for time period $embargo_{length}$, the total amount of time spent on unlocking the records would differ with regards to the amount of elapsed time, i.

If amount of time elapsed is less than the time-lock period $embargo_{length}$, then the repository would not yet contain any unlocked records. The amount of time required to unlock the records would then be the sum of the embargo period of the initial $R(0)$ number of records and the records added to R since $i = 0$. This summation can be represented as

$$R(0)(embargo_{length} - i) + \sum_{k=0}^{i-1} R_{update}(embargo_{length} - k), \quad embargo_{length} - i > 0 \tag{8}$$

If the amount of time elapsed is more than the embargo period, the initial $R(0)$ records would be unlocked, requiring computation on the time-locked records added to the repository since time i. An embargo period less than or equal to 0 refers to an unlocked record. Thus, the amount of computation time required to access these records would be

$$\sum_{k=0} embargo_{length} - 1 \; R_{update}(embargo_{length} - k), \quad embargo_{length} - i \leq 0 \tag{9}$$

Therefore, at time $i = 15$, the amount of computation time required to access all the records available in the repository, since $3 - 15 < 0$, $= r_1 + s_2 + t_3$, there will be $embargo_{length} = 3$ time-locked records, embargoed for time period 1, 2 and 3 respectively, which will require $1 + 2 + 3 = 6$ months of dedicated computation to be accessed.

Suppose that a harvester harvests data from the repository at harvest frequency of $H_{update} = 0.5$, the harvester harvesting records every $2\,i$ units. In case the repository dies and is inaccessible after a certain period of time, the harvester will have to perform dedicated computation on the local copy of records in order to unlock and access them; the amount of computation required can be calculated using the formulae above. The record updates published by the repository after the last harvest, as demonstrated in Figure 3, would be lost and cannot be recovered. Therefore, it would be preferable to coordinate harvesting from a repository on every repository update so as to avoid loss of any record updates and updated embargoed records, eventually reducing the computation time for accessing all the records in the repository.
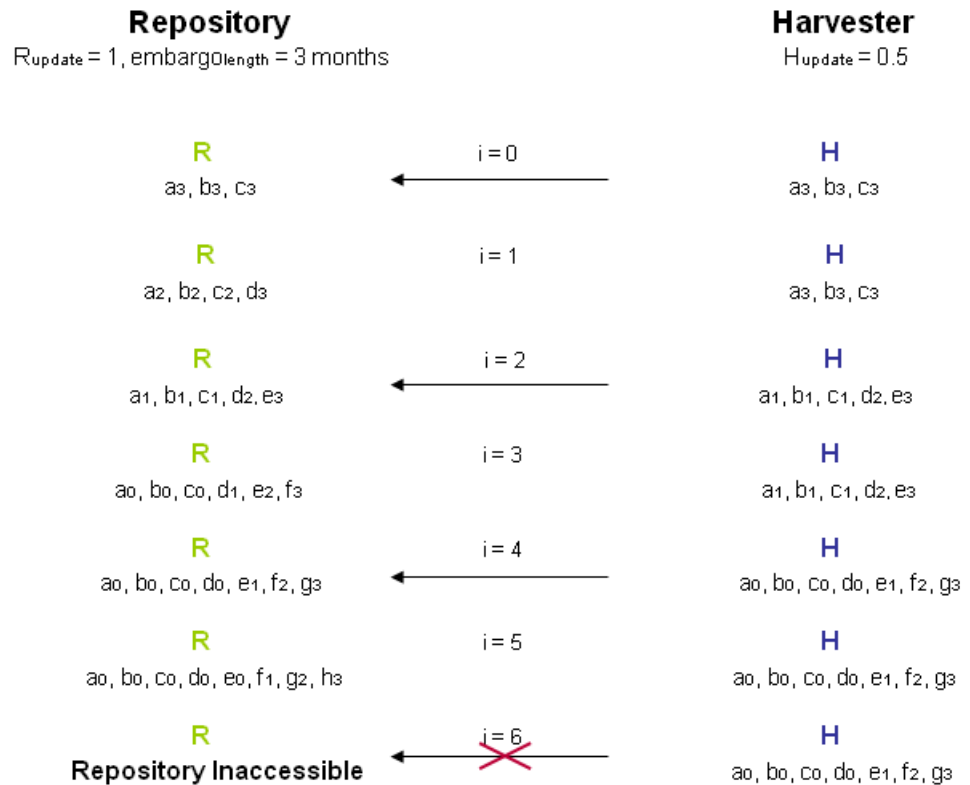
FIG. 3: Demonstration of the Preservation Risk Interval in an active repository. Repository begins with R(0) = 3 records, with $embargo_{length}$ = 3 months and R_{update} = 1 record. The harvester harvests data from the repository with H_{update} = 0.5 frequency.

IV.2 SOLUTION TO THE "PRESERVATION RISK INTERVAL"

With the emergence of large distributed systems, repositories have expanded to contain data pertaining to various access types. Embargoed records residing in these repositories need to be identified, encrypted and encapsulated for successful embargoed data dissemination. This section analyzes these aspects of preservation to formulate an effective and efficient solution that mitigates this preservation risk associated with embargoed content.

IV.2.1 EMBARGOED RECORD IDENTIFICATION

The repository example discussed here can be further analyzed to formulate an embargoed record identification method. The active repository contains records with publisher-imposed embargo time period of $embargo_{length}$ = 3 months, and the repository mandates $embargo_{decrement}$ = 3 updates of each embargoed record, then the embargoed records are updated with a weaker timelock every $embargo_{length}/embargo_{decrement}$ = 1 month. This results in a total of 3 instance updates. The embargo period inception of each embargoed record is from the publication, or creation date, which is $embargo_{start}$. Therefore, the end of the embargo period, $embargo_{end}$ for a record, can be easily projected by adding the embargo period $embargo_{length}$ to the datestamp $embargo_{start}$. A record under embargo can be identified by comparing the datestamp at the time of data dissemination with the publisher-imposed, global $publisher_{start}$ datestamp and the projected $embargo_{end}$ datestamp. If the record is identified as under embargo, then the remaining time of the embargo period of that record is calculated. A predetermined number of $embargo_{decrement}$ instance updates of the record, which is 3 in this example, with decreasing time-lock are provided. The remaining embargo period of the record is also used to calculate the number of instance update of the record, such as determining current embargo record version of 2 of 3 instance updates. This determines the complexity of the record encryption. Therefore, a decrease in the remaining embargo period results in the creation of a weaker encryption of the record that would require less computation time to break the encryption. Once the required embargo period has elapsed, an unencrypted instance of the record is created during content dissemination. For instance, if a record r_3 is published on Wed, 19 Sep 2007 09:58:19 GMT, the next update would be available after 1 month has elapsed, on Fri, 19 Oct 2007 09:58:19 GMT. An unlocked instance of the record would be available after the embargo period has elapsed, which would end on Wed, 19 Dec 2007 09:58:19 GMT. Algorithm 1 illustrates the embargo record identification process based on datestamp comparison of the record and the start of the embargo time period.

IV.2.2 EMBARGOED RECORD ENCRYPTION

The foremost design consideration during the formulation of an acceptable solution is the selection of a reliable cryptosystem that would ensure integrity of the embargoed

---

Algorithm 1 The Embargoed Record Identification process invoked during content dissemination

---

1: **for all** *records* **do**
2:    **if** $(embargo_{start} > publisher_{start})$ and $(embargo_{start} < embargo_{end})$ **then**
3:       calculate remaining $embargo_{length}$ for the record
4:       calculate record instance update
5:       calculate encryption complexity
6:       calculate new record datestamp
7:    **end if**
8: **end for**

---

content. A failed encryption system would result in loss of embargoed record security, and would thus fail to provide an acceptable solution. Timed-Release Encryption has been selected as the preferred encryption algorithm for time-locking embargoed content because of its mathematical properties. This system's properties and its application in this thesis research are further evaluated in Chapter V to establish its relevance and capabilities as a credible, and so far unbreakable, encryption algorithm.

IV.2.3   EMBARGOED RECORD ENCAPSULATION

The encrypted records residing in the active repository need to be accurately represented during data dissemination to facilitate embargoed record identification. Adequate information regarding the encryption method used needs to be included in the resulting document to facilitate record decryption and data verification upon decryption.

IV.3   SUMMARY

This chapter introduced the nomenclature that builds the foundation of the Preservation Risk Interval problem, and will be referred to for the rest of the problem discussion. It also illustrated this problem existing in a repository containing embargoed content. It also introduced the concept of a computation overhead involved in decrypting and accessing time-locked records in case a subsequent, unlocked version of the records is unavailable from the repository. It described the three essential components required to propose a viable solution aimed at mitigation of the "Preservation Risk Interval" of embargoed content. The next chapter introduces the existing

implementations of technologies that have been exploited to formulate the three components of the proposed thesis framework.

CHAPTER V

MITIGATION OF "PRESERVATION RISK INTERVAL" USING `MOD_OAI`

V.1 INTRODUCTION

The solution design considered in Section IV.2 has encouraged the formulation of the Time-Locked Embargo thesis model. This thesis model is a fusion of the time-locked puzzle concept with Apache module `mod_oai` and MPEG-21 DIDL complex object format for mitigation of preservation risk of embargoed content.

As described in Chapter II, the OAI-PMH metadata harvesting protocol has been incorporated in the `mod_oai` Apache module to provide an efficient content dissemination tool. `mod_oai` encapsulates base64 encoded data streams and related metadata in MPEG-21 DIDL XML format.

V.2 DESIGN CONSIDERATIONS

This thesis research is an extension of the existing `mod_oai` Apache module to include and handle content dissemination of embargoed content to service providers to facilitate more comprehensive content indexing and searching facilities. As a result, existing technologies have been modified and integrated for the implementation of this system. MPEG-21 DIDL preservation metadata schema, incorporated in `mod_oai` as the complex object encapsulation format, allows the flexibility required to extend the metadata tags to include additional information about the resource. As a result, this is the preferable complex object format used to encapsulate the embargoed records. It is easily adaptable without losing its interoperability property.

Disseminating embargoed content is possible only through encrypting the record. This encryption of embargoed records required a secure and reliable cryptosystem. RSA is the most widely used public-key cryptology algorithm, but has been successfully broken due to its mathematical properties. Therefore, the timed-release cryptosystem, a modification of the RSA algorithm with the additional property of being non-parallelizable, has been incorporated in the project design.

V.3   SYSTEM ARCHITECTURE

This thesis research is involved with the dynamic identification and subsequent encryption of embargoed records in a repository. Therefore, this research is focused on the identification of embargoed records in a repository and the calculation of the embargo period for which the record needs to be time-locked and encrypted before data encapsulation in the XML document. It also includes the MD5 [63] checksum of the data to validate integrity of a decrypted instance of the record.

In response to a GetRecord or ListRecords `mod_oai` HTTP request with metadata prefix oai_didl, the embargoed identification, encryption and encapsulation components incorporated in the Apache module are invoked. Algorithm 2 describes the sequence and interaction of these components.

Algorithm 2 Interaction of various record data and metadata creation components

---

```
 1: if (Request Verb = ListRecords) or (Request Verb = GetRecord) then
 2:     Records Index Creation
 3:     for all records do
 4:         Preservation Metadata Creation
 5:         Metadata Encapsulation in MPEG-21 DIDL data item
 6:         if record is under Embargo then
 7:             Dynamic Embargoed Record Identification
 8:             Record Datestamp modification
 9:             Dynamic Embargoed Record Encryption
10:             Dynamic Embargoed Record Encapsulation
11:         end if
12:         if Include Resource by-value then
13:             Resource Encapsulation in MPEG-21 DIDL component
14:             if record is under Embargo then
15:                 Resource MD5 hash Encapsulation in MPEG-21 DIDL data descriptor
16:                 Puzzle Parameters Encapsulation in MPEG-21 DIDL data descriptor
17:             end if
18:         else
19:             Include Resource reference in MPEG-21 DIDL component
20:         end if
21:     end for
22: end if
```

The functionality of these components is further described in detail to enunciate the creation of the resulting XML document representing a record under embargo.

V.3.1 DYNAMIC EMBARGOED RECORD IDENTIFICATION AL-GORITHM

The general solution for embargoed record identification discussed in section IV.2 has been further developed and applied in this research. Algorithm 3 is the algorithm of the program that has been incorporated in `mod_oai` to identify records that are under embargo.

Algorithm 3 Dynamic Embargoed Record Identification Process incorporated within `mod_oai`

1: $publisher_{start}$ = Global Zulu date when embargo period for each record begins
2: $embargo_{length}$ = 365 {embargo time period for each record in days}
3: $embargo_{decrement}$ = 12 {number of embargoed record updates}
4: $lockStart$ = startDate in unix seconds
5: $lockDuration$ = $embargo_{length}$ in seconds
6: **for all** $records$ **do**
7:     $currentTime$ = current date in unix seconds {input current date from the system}
8:     $fileTime$ = current record modified date,$embargo_{start}$, in unix seconds
9:     **if** ($fileTime < lockStart$) or ($fileTime \leq (currentTime - lockDuration)$) **then**
10:       the record is not under embargo
11:     **else**
12:       **if** ($(fileTime + lockDuration) > currentTime$) **then**
13:         $noOfSecsInInterval = (lockDuration/p)$
14:         $elapsedTimeFraction = (currentTime - fileTime/lockDuration)$
15:         $intervalNo = elapsedTimeFraction$ * p
16:         $elapsedLockTime = intervalNo$ * $noOfSecsInInterval$
17:         $intervalsLeft = p$ - $intervalNo$
18:         $lockTimeLeft = intervalsLeft$ * $noOfSecsInInterval$
          {lockTimeLeft is used to linearly interpolate the anticipated computation timeUnit}
          {calculate new timestamp of the file according to last interval update}
19:         $newTimestamp = elapsedLockTime + fileTime$ {calculating next update}
20:         $nextUpdate = (intervalNo + 1)$ * $noOfSecsInInterval$
21:         $nextTimestamp = nextUpdate + fileTime$
22:         convert $newTimestamp$ and $nextTimestamp$ integer variables from seconds to Zulu time
23:         print $intervalNo$ as the current version of the record, out of a total of $embargo_{decrement}$ versions
24:         print the $nextTimestamp$ in Zulu time as the next anticipated update timestamp
25:         print $lockTimeLeft$ as the anticipated computation time required to unlock the time-locked puzzle
26:       **end if**
27:     **end if**
28: **end for**

The publisher-imposed start date for the embargo period of each record has been included in the `mod_oai` configuration file as a variable that can be modified and set by the user accordingly. The duration of the embargo period, with a granularity of days, has also been established as a known variable. The number of instance

updates, or intervals during these updates, is also a known variable and is included in the configuration file as required input. These variables can be adjusted to modify the effective embargo time period for the records and the number of instance updates desired for each record under embargo. A sample `mod_oai` configuration file can be found in Section A.1 of Appendix IX. The above algorithm and remaining examples of embargoed record identification and encryption are based upon an $embargo_{length}$ = 365 days, with $embargo_{decrement}$ = 12 instance updates for each record. A later version of each record would correspond to a less complex time-lock puzzle that would require less computation time to effectively break the encryption and access an unencrypted copy of the record.

The proposed algorithm computes time values in unix seconds to enable mathematical manipulation on time values and then converts the time in seconds back to ISO 8061 time before being included in XML output. After identifying a record as under embargo, the algorithm first calculates the time fraction that has elapsed since the start of the embargo period. This fraction is then converted into an integer value that represents the current number of the instance update. This interval number is then used to determine the effective remaining lock-time, on which the complexity of the time-lock puzzle is dependent.

The remaining lock-time in seconds is the computation time it should take to access the record once it has been time-locked. The timed-release cryptology algorithm requires an integer input value that determines the complexity of the record encryption. Thus, various tests, described during system evaluation in Chapter VI, have been conducted to linearly interpolate and map the remaining lock-time with an appropriate integer value that is used as the input for the embargoed record encryption algorithm. Figure 4 provides an example listing the mapping of about 38 hours of remaining lock-time with a puzzle complexity value of 80.

V.3.2 DYNAMIC EMBARGOED RECORD ENCRYPTION

The MIT/LCS timed-release cryptosystem, with minor modifications, has been incorporated in `mod_oai` for embargoed record encryption. This section first describes the original timed-release puzzle algorithm and later enunciates the modified version incorporated in this thesis research.

The LCS35 Time Capsule Crypto-Puzzle [64], created in 1999, has been designed
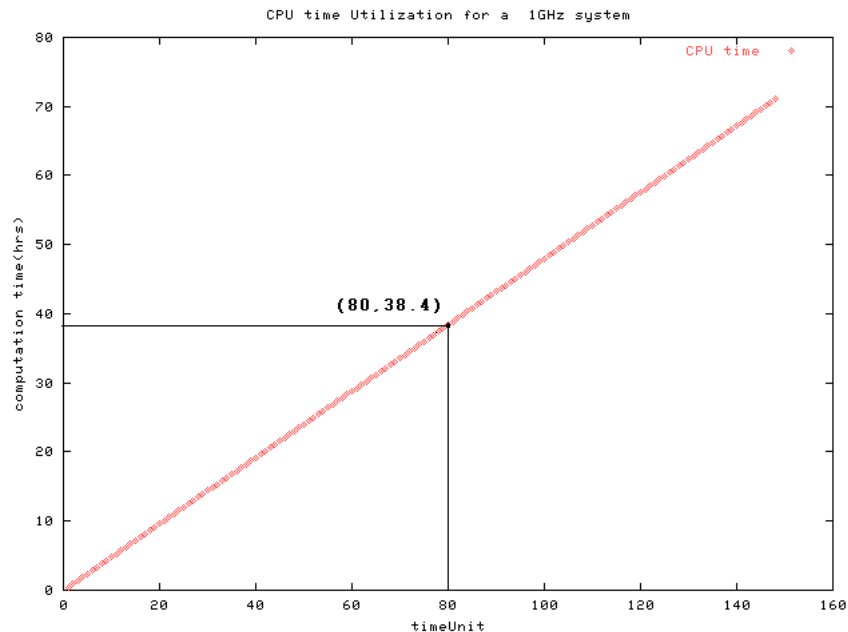
FIG. 4: Linear mapping of remaining locktime with puzzle complexity value

to take about 35 years of linear computation to solve. The puzzle gives due consideration to the fact that during the upcoming years, improvement in processing speed that will inevitably reduce the computation time required to break the puzzle. According to Moore's Law [44], the number of transistors on an integrated circuit increases exponentially, doubling approximately every two years. Therefore, due to the large 35-year embargo period of the puzzle, Moore's Law has been considered while calculating the complexity of the puzzle and the t puzzle complexity value in turn. The puzzle follows the future trend described by SEMATECH National Technology Roadmap for Semiconductors [19] that predicts an exponential increase in processing speed by a factor of 13 from 1999 until 2012. A further increase in speed by a factor of 5 until 2034 has been estimated and taken into account in the algorithm. In order to ensure the computation time of 35 years, it is assumed that a faster computer will be used every year to perform the required sequential computation to break the LCS35 puzzle.

Algorithm 4 has been included in `mod_oai` with a few, required modifications. Upon receiving a `mod_oai` request, if the record is identified as under embargo, the

---

Algorithm 4 MIT/LCS35 time-lock puzzle creation

1: $squaringsPerSecond = 3000$ {compute number of squarings to do each year}
2: $secondsPerYear = 31536000$
3: $squaringsFirstYear = secondsPerYear * squaringsPerSecond$
4: $years = 35$
5: $t = 0$
6: $s = squaringsFirstYear$
7: **for** $i = 1999$ till $i \leq 1998 + years$ **do**
8:     $t = t + s$ {do s squarings in year i}
       {apply Moore's Law to get number of squarings to do next year}
9:     $growth = 12204$ { x13 up to 2012, at constant rate}
10:    **if** $i > 2012$ **then**
11:       $growth = 10750$ { x5 up to 2034, at constant rate}
12:    **end if**
13:    $s = (s * growth)/10000$
14: **end for**
15: print squarings (total) as t
16: print Ratio of total to first $year = t/squaringsFirstYear$ {generating RSA parameters}
17: $primelength = 1024$
18: $twoPower = $ shift left primelength(1)
19: $prand = $ large random integer for prime p seed input by user
20: $qrand = $ large random integer for prime q seed input by user
21: $p = 5$
22: $q = 5$ { 5 has maximal order modulo 2^k (see Knuth)}
23: $p = (p^{prand})$ mod $twoPower$
24: $p = $ get next prime of p
25: $q = (q^{qrand})$ mod $twoPower$
26: $q = $ get next prime of q
27: $n = p * q$
28: $pMinus1 = p - 1$
29: $qMinus1 = q - 1$
30: $phi = pMinus1 * qMinus1$ {Generating final puzzle value w}
31: $u = (2^t)$ mod phi
32: $w = (2^u)$ mod n {obtain and encrypt the secret message}
33: $sgen = $ the string for the secret {append seed for p as a check}
34: $sgen = sgen + $ seed value b for p is $prand$ {Base256 interpretation of the given string sgen}
       {convert character of sgen into ascii equivalent integer value}
35: **for** $i = 0$ till $i¡$ length of sgen **do**
36:    $c = sgen[i]$
37:    $secret = $ shift left$(secret)$
38:    $secret = secret + c$
39: **end for**
40: $z = (secret)$xor(w) {print puzzle parameters in output file}
41: print n and t parameters
42: print final puzzle z
43: print "To solve the puzzle, first compute $w = 2^{(2^t)}(mod\, n)$. Then exclusive-or the result with
       z. The result is the secret message (8 bits per character), including information that will allow
       you to factor n. (The extra information is a seed value b, such that $5^b(mod\, 2^{1024})$ is just below
       a prime factor of n.)"

---

following modified Algorithm 5 is called to dynamically time-lock the file during data dissemination. It takes the variable *timeUnit* as input and outputs the time-locked instance of the record along with the puzzle variables n and t required to break the puzzle, along with the instructions on how to break the puzzle.

Algorithm 5 Dynamic Embargoed Record Encryption algorithm incorporated within `mod_oai`

1: $squaringsPerTimeunit = 3000$
2: $secondsPerTimeunit = 1800$
3: $squaringsFirstTimeunit = secondsPerTimeunit * squaringsPerTimeunit$
4: $t = 0$
5: **for** $i = 1$ till $i \leq timeUnit$ **do**
6:     $t = t + squaringsFirstTimeunit$ {the number of squarings depends on timeUnit. t increases linearly}
7: **end for**
8: $primelength = 1024$ {generating RSA parameters}
9: $twoPower = $ shift left primelength(1)
10: $prand = 3$
11: $qrand = 5$
12: $p = 5$
13: $q = 5$ {5 has maximal order modulo 2^k (see Knuth)}
14: $p = (p^{prand})$ mod $twoPower$
15: $p = $ get next prime of p
16: $q = (q^{qrand})$ mod $twoPower$
17: $q = $ get next prime of q
18: $n = p * q$
19: $pMinus1 = p - 1$
20: $qMinus1 = q - 1$
21: $phi = pMinus1 * qMinus1$
22: $u = (2^t)$ mod phi {Generating final puzzle value w}
23: $w = (2^u)$ mod n {convert the file in Base256}
24: **while** $buffer = $ read a char from file **do**
25:     $c = buffer$ {convert character into ascii equivalent integer value}
26:     $secret = $ shift left$(secret)$
27:     $secret = secret + c$
28: **end while**
29: $z = (secret)$xor(w) {print puzzle parameters}
30: print n and t parameters in a string variable
31: print extra information required to break the encryption
32: Base64_encode(z)
33: print the puzzle parameters and Base64 encoded z in XML document

---

Algorithm 5 has been programmed in the C programming language and incorporated in `mod_oai`. The GMP C library[1] has been used to declare the large numbers produced in the algorithm. The logic of the timed-release cryptographic algorithm has been preserved. It has been amended to accept binary files as input, to permit

---

[1]http://gmplib.org/

all files to be accessed and encrypted irrespective of their MIME type. To allow files with various MIME types to be time-locked, it was not possible to append or include the seed value b in the file during data encryption. The MD5 cryptographic hash function has been used to create 32-character hash values of the file to verify the integrity of the file contents upon decryption.

Moore's Law has been abandoned in favor of linear increase in puzzle complexity. This is due to the considerably smaller time-lock period, $embargo_{length} = 365$ days, of the files under embargo. Since the files under consideration are under a temporary embargo, and will be unlocked and accessible free of cost after the embargo period has elapsed, it was deemed unnecessary to consider the trend in computation speed during calculations. If a file is under embargo for two years, it will require approximately two years of linear computation on an average computer to unlock. In order to take advantage of increasing technological speed, a year and a half can be spent in idle time, and then a faster computer can be utilized to perform necessary computation to unlock the encryption in the remaining six months. Due to the linear growth in computation speed, the time required to wait for faster processors to be built and then used for computation has to be considered within the required time to unlock the file. The total amount of time dedicated to computation is determined by the speed of the processors used. Therefore, spending the entire embargo period in continual computation would be equivalent to the idle and subsequent computation time spent to effectively break the lock.

V.3.3   DYNAMIC EMBARGOED RECORD ENCAPSULATION

The resulting encrypted instance of the record under embargo is accurately represented and encapsulated in a resulting MPEG-21 DIDL XML document. This document reflects the appropriate changes in the included metadata to be identified as an encrypted instance of the record. The resulting XML document to a `mod_oai` *GetRecord* request has been included in Section B.1 of Appendix IX. Figure 5 is the resulting DIDL document structure that encapsulates data and metadata pertaining to a record under embargo.

The two occurrences of the last-modified datestamp of the record have been updated in the resulting XML document to reflect the latest embargoed record instance. This datestamp represents the last instance when the record was encrypted with the latest embargo period. It has been updated in the DIDL header section, in ISO
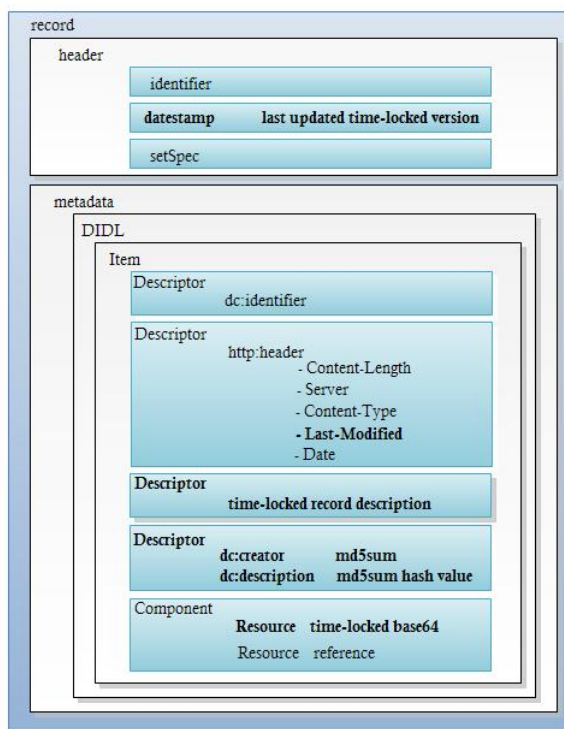
FIG. 5: MPEG-21 DIDL structure of an embargoed record

8061 [33] date format, and in the http metadata descriptor, in RFC 822 [10] date format. The resource representation of the record has been updated; if the file size is moderate, the document would contain a base64ed instance of the time-locked record. The resource reference would also be modified and redirected so that an encrypted copy of the record is returned to the user upon dereferencing. The MD5 checksum of an unlocked instance of the record has been included in a descriptor in the response to provide sufficient information for integrity verification upon record decryption. It has been encapsulated in a DIDL descriptor element. This checksum can be substituted for any favorable cryptographic hash function for data validity of an unlocked instance of the record.

An extra descriptor DIDL element has been introduced that identifies the record as time-locked and encapsulates related information such as the instance of the embargoed record update, the original start of the record embargo, and when an unlocked instance of the record can be anticipated. It also provides the variables and method to break and access the encrypted record, and includes information on the

estimated computation hours required to achieve this task.

INCLUDING RESOURCES BY REFERENCE

The `mod_oai` Apache module allows the record resource to be included in the XML document by reference as well as by value. The reference URL pertaining to an encrypted record has been modified to be directed through a script that allows the file accessed to be time-locked before data display. It provides the same information included in the DIDL document required to unlock and access the encrypted record instance. Following is the resulting HTML document by accessing the record via reference.

> This version of the record is 7 of 12 separate encryptions, each of which is successively easier to break.
>
> It will take approximately 3650 hours of computation to break this time-lock.
>
> The next update will be available on 2008-01-16T20:56:15Z.
>
> Crypto-Puzzle for LCS35 Time Capsule.
>
> Puzzle parameters (all in decimal): $n = 398399$ $t = 264600000$.
>
> $z =$
>
> 31323917451802555277390938846180173530238875932282538037348989337556205685991477714451887948857360790660493418675968261881575537176434252284573461416955038100854208006858408848533030529112351035806118745475819960808730529812728678278348234778837646499326132378882478546932929288832921937860582941559740180707580176848638298208892890857685642513650264362418878974023890672963063921156510954869322578645204486019792177839161968131852251600257117838822963932415158001790747932504269325051680296385145617169875420464883115780691334411515936834622658540367888367276379634776192539852572581551787878787389507779350721179436015038787973434062174205481417757375084090152723132539913075822432942722392338647890068064991041419833989430494142355423188103584627664110239652311557235465679119392364041970374784288899755676142220795878085575830515477502779988607514835114229142479340931233616657201906335470658288113526360375446239476156339813286886592959666859530138724095220527363673590956280610867592451994241073600119108613641282714292009799835549500827514545511071440172673092781119493704018751089383336458839064091102274635368029685755341573783494274696186747975923748316080527772530708563880243765670169430160602877418215143129006885672412173355340099700662224222324807663760631601449721071360342376183475553638339443195855670

79481331214780889879688072045098643059528284602520962734700112787090145761686666403078647807907212920481649562418788180556860951650796784105600966390776531550670233117972202526138055202246514752070767199375489918275462357311229650860107262853324746521749026524626264260057977192138940136536360818953201181350296738984435407988890427745373128667752398902803688766439445119156262271776380811877237296225401342555096194930699304161717824997634765671816700830261089715949109048874874874243703017189418499622867183451181300980365140915007212326120861

To solve the puzzle, first compute $w = 2^{(2^t)} \pmod{n}$.

Then exclusive-or the result with z.

(Right-justify the two strings first).

The result is the secret message (8 bits per character).

The resource URL is a reference to an HTML webpage that, upon access, displays the resource in decimal values without base64 encoding, which is the original output of the timed-release puzzle. The returned page iterates the information included in the XML document that is required to decrypt the accessed encrypted record.

V.4   SELECTING APPROPRIATE VALUES OF T

The timed-release cryptography algorithm has been designed to take an integer variable $timeUnit$ as an input. This $timeUnit$ value linearly increases the complexity of the puzzle by increasing the t value in the 2^{2^t} computation performed to break the file time-lock. A puzzle corresponding to a higher t value requires more time to break the encryption. This decryption time can be mapped with $embargo_{length}$, the remaining embargo period of the file that was calculated during dynamic embargoed record identification. Due to the mathematical properties of this cryptography method, the time required to break the time-lock puzzle is directly dependent upon the speed of the processor used to perform the required calculations. An appropriate value of $timeUnit$ can be selected to accurately calculate the puzzle complexity and the time required to break and access the encryption in relation to the desired processor speed.

A correlation between the remaining embargo period, $embargo_{length}$, and the t value on a particular computer system can be calculated by assimilating the results of the timed-release algorithm. To compute this correlation, the timed-release cryptography algorithm has been executed with increasing values of $timeUnit$ to create

puzzles of increasing complexity. These puzzles have then been broken to determine the amount of time required to break the time-lock and subsequently access the encrypted content. A table of increasing $timeUnit$ values and their corresponding decryption time can be created to describe this dependency. This unlock time can then be mapped to $embargo_{length}$, the remaining embargo period for a record, to ensure that the decryption time is no less than $embargo_{length}$.

This linear proportion can be described as

$$embargo_{length} \propto tU \tag{10}$$

where $embargo_{length}$ is the remaining embargo period for a record, and tU is the corresponding timeUnit value used by the timed-release algorithm to create the time-lock puzzle.

This algorithm can be executed on a different processor to compute the linear correlation between $embargo_{length}$ and tU pertaining to the system speed. During this thesis research, various experiments have been conducted on a variety of systems to demonstrate a linear correlation between $embargo_{length}$ and tU, which are later described during the evaluation of this approach in the next chapter.

V.5   SUMMARY

This chapter introduced in detail the implementation method of the solution to the "Preservation Risk Interval". It described the design considerations, and the system architecture of embargoed data identification, encryption and encapsulation implemented in this model. It described how the complexity of the time-lock puzzle can be personalized for the computer system being utilized during implementation to ensure data integrity and security. This implementation method is further evaluated in Chapter VI.

CHAPTER VI

SYSTEM EVALUATION

This chapter examines how the system described in the previous chapter has been evaluated to ensure that embargoed content can be successfully time-locked and disseminated in `mod_oai` upon an OAI-PMH request. Various tests have been conducted to ensure that this system satisfies the requirement of successfully disseminating embargoed content with an acceptable time overhead, demonstrating the ability to mitigate the Preservation Risk Interval of embargoed content.

The proposed system has been evaluated on two criteria: to ensure that the time required to unlock and break the time-lock puzzle is equivalent to the embargo time period, and that this embargoed content is successfully disseminated via `mod_oai` with minimum time overhead.

The timed-release cryptography system is dependent upon the computation speed of the machine utilized to eventually break the time-locked puzzle. The linear correlation between $embargo_{length}$, the unlock time required, and tU, the timeUnit value, to be used for record encryption can be described as

$$f(x) = \frac{embargo_{length}}{tU} \tag{11}$$

where x is the processor speed of the machine utilized for computation. An increase in processor speed x would result in a decrease in $embrgo_{length}$. Puzzles have been created using this timed-release algorithm on various computer systems to establish a linear correlation between $embargo_{length}$ and tU, and to determine and analyze this variant $f(x)$ value differing with each system speed x.

The proposed system has also been tested by harvesting a website with no time-locked data and with entire website content as time-locked to demonstrate the feasibility of this approach and to establish an estimated time overhead, dependent upon the content-size to be encrypted, before effective data dissemination.

VI.1 EFFECT OF COMPUTATION SPEED ON $EMBARGO_{LENGTH}$

The timed-release algorithm has been executed on machines with varying speeds x, in order to determine the effects of processor speed on $embargo_{length}$, the time required to decrypt and access a time-locked record.

Four x values have been empirically calculated from known data in the course of this research. A Sun Solaris cluster comprising of 31 nodes has been utilized for performance testing during the research. 26 nodes of this cluster have the processing speed of $x = 1.6$ GHz, and 5 nodes have a processing speed of $x = 1.8$ GHz. Two machines with a computation speed of $x = 1$ GHz and $x = 0.75$ GHz respectively, have also been utilized to compute and compare the time required to break the timed-release cryptography puzzle.

An identical text file was used by these machines as input to create puzzles of varying complexity by using the timed-release algorithm, starting with the input value $timeUnit = 1$. This timeUnit value is used by the timed-release algorithm, as described in section V.3.2, to calculate the t value used for data encryption, which remains unchanged throughout the tests performed. As described in the timed-release algorithm, the input $timeUnit$, or tU, value linearly increases the puzzle t value. Each increment in the $timeUnit$ value increases the t value by a value of 5,400,000. Therefore, tU, the timeUnit value used as input in the algorithm, can be mapped on to the t value as

$$t = 5400000tU \tag{12}$$

The created time-lock puzzles of varying complexity are then broken to record the amount of decryption time on the particular system. The accumulated values of effective $timeUnit$ values and unlock time $embargo_{length}$ have been plotted to demonstrate a linear correlation between puzzles with increasing complexity and unlock time $embargo_{length}$. All time values have been recorded in seconds time granularity.

Figure 6 is a plot of the datapoints gathered by tests performed on the four classes of machines that demonstrate the relationship between $embargo_{length}$ and tU.

As demonstrated by Figure 6, the unlock computation time, $embargo_{length}$, increases linearly with increasing $timeUnit$ puzzle complexity. As discerned from equation 11, x, the factor of puzzle complexity, is the rate of increase of this unlock time with respect to timeUnit tU on a particular system. This x factor decreases with an increase in computation speed of the system used.

Results demonstrate that a decrease in computation time in relation to an increase in processing speed is consistent across various x classes of machines. Table IV contains the corresponding $f(x)$ value of these x class of machines, which is the slope of the four classes of machines, utilized to determine the correlation between tU and $embargo_{length}$.

53
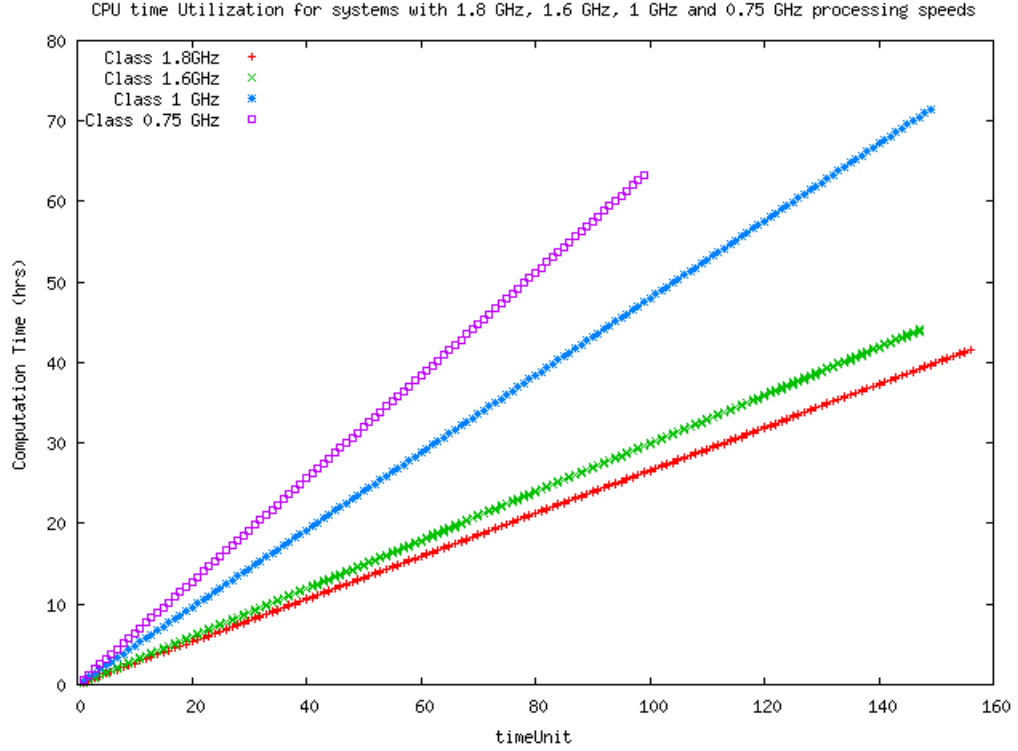
FIG. 6: Unlock time of a time-locked puzzle with increasing complexity on four classes of machines

From the graph in Figure 6 and the corresponding *f(x)* values of the four x classes of machines in Table IV, it can be discerned that the rate of increase of a particular class of machine can be projected from any one selected class of machine. An increase in the rate of change $f(x)$ for a given class of machine is inversely proportional to the computation speed x of that class; that is,

$$x \propto \frac{1}{f(x)} \tag{13}$$

As the processing speed of the machines increases, the slope of the linear graph corresponding to that speed decreases. Therefore, for any one class of machine, three *f(x)* values for that particular computation speed can be extrapolated using the *f(x)* values of the rest of the three classes of machines.

As illustrated in Table IV, for class $x = 0.75$ GHz processing speed, the actual, calculated rate of change of $embargo_{length}$ with respect to an increase in tU, $f(x)$, is

x Class (GHz)	$f(x)$ value
0.75	2302.16
1	1727.61
1.6	1079.14
1.8	959.78

TABLE IV: Corresponding $f(x)$ values of the four x classes of machines.

2302.16. Since the relationship between x and $f(x)$ is inversely proportional, it can be mathematically described as

$$0.75 \propto \frac{1}{2302.16} \tag{14}$$

With the known pair of values for $x = 0.75$, the $f(x)$ value of a 1GHz machine can be projected using this inverse proportion relation as

$$\frac{0.75}{1} = \frac{f(1)}{2302.16} \tag{15}$$

$$f(1) = 0.75 * 2302.16 \tag{16}$$

$$f(1) = 1726.62 \tag{17}$$

Thus, an $f(x)$ value for a class of machine can be projected using the equation

$$\frac{x_j}{x_k} = \frac{f(x_k)}{f(x_j)} \tag{18}$$

The $f(x_k)$ value corresponding to the known x_k computation speed can be projected with the formula

$$f(x_k) = \frac{f(x_j) * x_j}{x_k} \tag{19}$$

A calculated $f(x)$ value for a particular class of machine that is closer to the actual, calculated $f(x)$ value in Table IV indicates the observance of Moore's Law, whereby concluding that this inverse correlation can be utilized to project the $f(x)$ value for various classes of machines. Table V lists the projected $f(x)$ values for each

	class 0.75	class 1	class 1.6	class 1.8
class 0.75	**2306.16**	2303.48	2303.17	2303.47
class 1	1729.62	**1727.61**	1726.62	1727.60
class 1.6	1081.01	1079.76	**1079.14**	1079.75
class 1.8	960.90	959.78	959.24	**959.78**

TABLE V: Projected $f(x)$ slopes for x classes of machines. Bold values are real slopes.

of the four classes. The bold values correspond to the real *f(x)* values resulting from the performed computational tests.

It can be discerned from the above results that a projected slope of unlock time and timeUnit with respect to a particular processing speed is a realistic and reliable assessment. This projected slope can be further applied to calculate the timeUnit tU value that should be used to time-lock a file for a specified lock time.

For instance, if a file is to be locked for 2 years on a 2.5 GHz machine, an appropriate *f(x)* value corresponding to this processing speed x can be calculated. This *f(x)* value is the rate of change for unlock time with respect to increasing values of time Unit and can be represented as equation 11. Substituting *f(x)* with its calculation in equation 19 gives us the new formula for determining unlock time embargo$_{length}$,

$$embargo_{length} = \frac{f(x_j) * x_j}{x} tU \tag{20}$$

where $f(x_j)$ and x_j are a known *x-f(x)* value pair and x is the processing speed for which a rate of change is to be projected. With a known unlock time value, equivalent to *embargo$_{length}$*, a corresponding *timeUnit* value to be used as input in the timed-release cryptography algorithm can be calculated as

$$tU = \frac{x}{f(x_j) * x_j} embargo_{length} \tag{21}$$

Substituting the class 1 value as x_j, and its calculated $f(x_j)$ value would result in the formula

$$tU = \frac{x}{1727.61} embargo_{length} \tag{22}$$

that can be used as a benchmark to project the timeUnit value for the desired class of machine with a known x. This formula can be utilized to calculate tU for known $embargo_{length}$ on the machine selected for data encryption during data dissemination via `mod_oai`.

Thus, the tU value required to time-lock a file for $embargo_{length} = 2$ years, which is 63115200 seconds, on machine $x = 2.5$ GHz can be calculated as

$$tU = \frac{2.5}{1727.61} 63115200 \tag{23}$$

$$tU = 9133 \tag{24}$$

VI.2 EXPERIMENTAL EVALUATION

A website containing this thesis research data has been utilized as the experimental website that is harvested for `mod_oai` performance testing. This data collection contains 525 files and comprises of 17.3 MB of data. 63% of the files are text files of various sizes, and the average size of each file is approximately 33 KB.

Figure 7 is a graphical representation of the size of each file in relation with the number of files comprising the website. The file size (y-axis) in this graph has been represented in log scale.This data collection has not been created specifically for performance testing and is the actual data collected during this thesis research. This collection has been utilized to simulate a real website where the collection may be varied and asymmetrical in size.

During experimentation, harvest times via `mod_oai` of this data conglomerate have been collected without data encryption, and with a desired time-lock encryption, $embargo_{length}$, of one year. This $embargo_{length}$ can be modified according to the embargo length of the records contained in the repository. A variable $modoai\_encode\_size$, contained in the module's configuration file, determines the maximum file size that is allowed to be included by-value in the resulting XML document. It subsequentially controls the size of each resulting XML document during data dissemination. Therefore, differing harvest times of the website with increasing values of $modoai\_encode\_size$ have been collected to analyze the performance of `mod_oai` with increasing quantity of time-locked data in the resulting XML document in response to a `mod_oai` *ListRecords* request. The number of total XML document responses required to harvest the entire contents of the website have also been included for further comprehension.
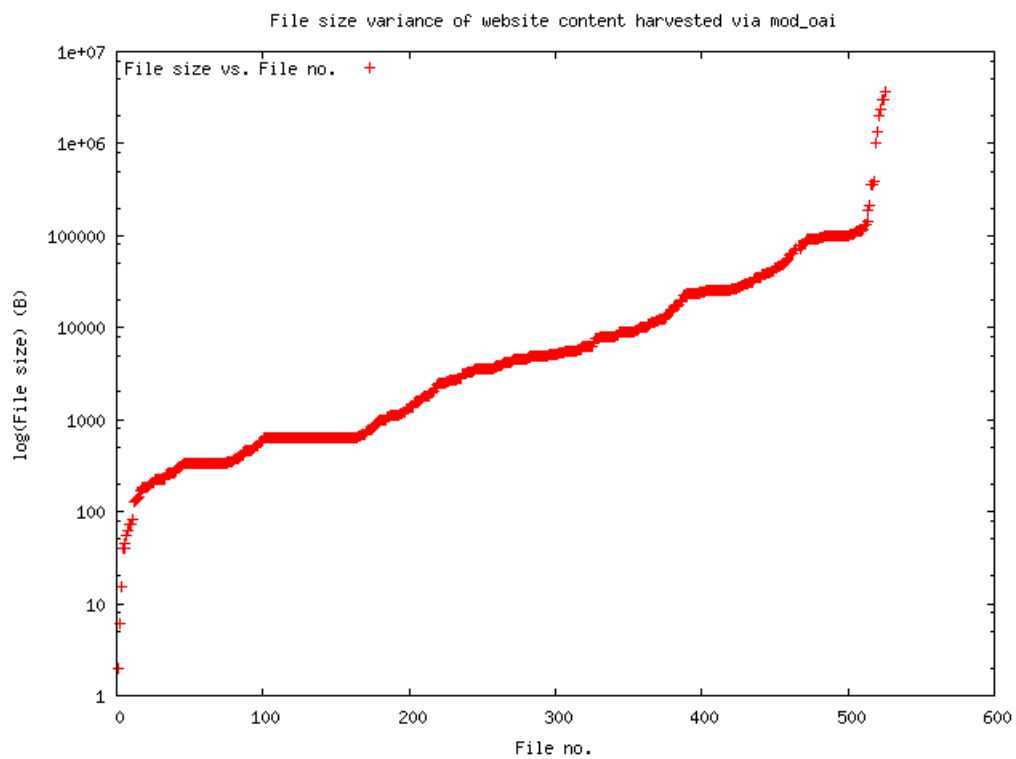
FIG. 7: File size variance of website content harvested during mod_oai performance testing

modoai_encode_size (Bytes)	No. of Responses	None Time-locked (sec)	All Time-locked (sec)
150,000	69	16.2	555
300,000	35	9.5	635
500,000	24	7.6	913
700,000	16	5.4	988
1,000,000	13	4.5	937
5,000,000	6	4.5	3648
10,000,000	6	4.5	10380
15,000,000	6	4.7	10962

TABLE VI: Wallclock harvest times of website with varied *modoai_encode_size* and embargoed content.

As demonstrated by Table VI, an increase in the *modoai_encode_size* results in a larger XML document response and fewer number of total XML document responses to a *ListRecords* request. An increase in *modoai_encode_size* results in an increase in the number of files included by-value via base64 data inclusion in the data harvest. With no data under embargo, there is no lock-time overhead, and the increase in the *modoai_encode_size* favors the increase in by-value data inclusion and results in faster data dissemination.

With the entire website content under embargo, an increase in *modoai_encode_size* leads to an increase in the harvest time due to the time overhead required to lock the increasing volume of data to be included in the response as base64 datastream. The entire website content is harvested as a base64 encoded datastream in 3.2 hours, with *modoai_encode_size* set to 10 MB, which is the optimal performance time of embargoed data harvest. At *modoai_encode_size* = 15 MB, an overhead time of exporting very large XML document responses is included as penalty in the total harvest time.

A graphical comparison of log representations of embargoed and unembargoed website harvest times with increasing *modoai_encode_size* is illustrated in Figure 8.

An increase in by-value content inclusion during content dissemination leads to an exponential increase in harvest time. The file contents of the website were individually time-locked, with embargo time-period $embargo_{length}$ = one year, to determine that the amount of time required to time-lock a file is dependent upon the size of
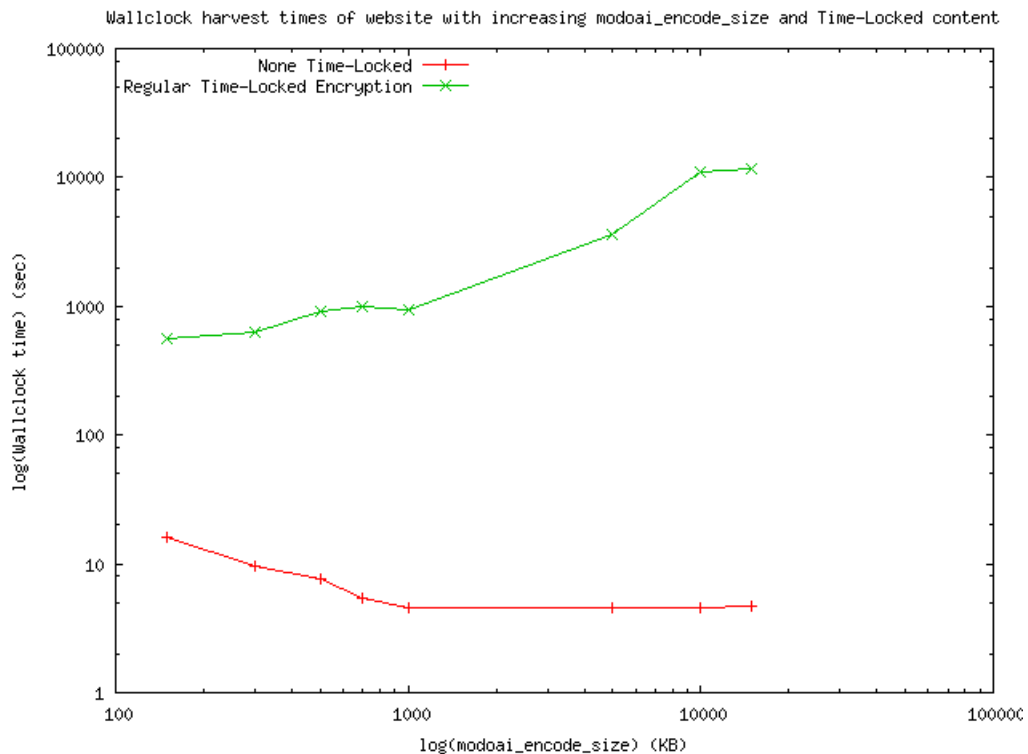
FIG. 8: Comparison of harvest times between unlocked and time-locked website content during dynamic data dissemination

the file. Figure 9 is the resulting graph representing the time required to time-lock individual files, in log representation, in relation to their respective file size.

The tests were repeated with differing embargo periods l to ensure that the amount of time required to time-lock a file is independent of the embargo period l, as can be discerned from the time-lock puzzle code included in Section V.3.2. The time required to time-lock a file increases exponentially in relation to the file size at a rate of $O(n^2)$. This approach of dynamic data dissemination of embargoed content becomes infeasible with large file sizes.

VI.3   SUMMARY

This chapter has evaluated and examined two significant aspects of the proposed system. The amount of time required to decrypt and access a time-lock puzzle should not be less than the expected embargo period $embargo_{length}$. Various tests
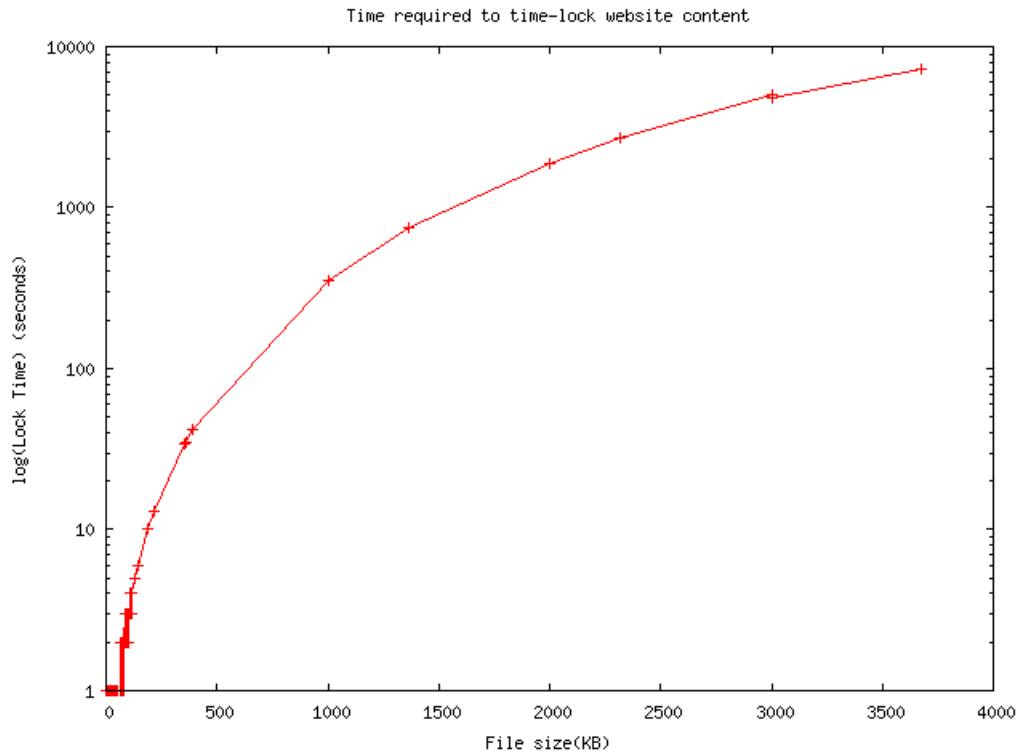
FIG. 9: Time required to individually time-lock files contained in the test website

have been performed to determine that the amount of time required to decrypt and subsequentially access an unlocked instance of the record is dependent upon the computation speed of the system utilized for data decryption. Therefore, the *timeUnit* variable during the creation of the time-lock puzzle can be modified to change the puzzle complexity to ensure that the puzzle cannot be broken before a pre-determined amount of computation time has elapsed on the utilized machine.

An evaluation of performance tests concludes that the overhead time for including encrypted embargoed content during dynamic data dissemination upon a `mod_oai` request is quite large. This is due to the $O(n^2)$ time required to time-lock a file, with n being the size of the file, rendering this approach of embargoed data encapsulation inefficient. An optimization of this harvest time is explored in the next chapter.

CHAPTER VII

OPTIMIZATION WITH CHUNKED ENCRYPTION

An evaluation of the system developed thus far reveals that that time required to encrypt a file during dynamic data dissemination upon a data request is too large, rendering this approach infeasible. This chapter introduces the method of time-locking files using "chunked" encryption, resulting in system optimization by a factor of 70. It also describes the necessary modifications rendered to the DIDL document structure to encompass this optimization.

VII.1   CHUNKED DATA ENCRYPTION

Upon analysis of the effect of increasing file size on encryption time, as demonstrated in Figure 9, and the discovery of the $O(n^2)$ property of time required to time-lock content, it can be intrinsically determined that the size of file has a significant impact on the data dissemination time. A file of size 100 KB on a 1.8 GHz machine requires 3 seconds to be time-locked, whereas a file with double the size of 200 KB requires 13 seconds. If the 200KB file is divided into two chunks of 100 KB each, the 200 KB file can be time-locked, without parallelism, in 6 seconds, resulting in a 54% improvement. Let x be a file of size 200 KB, and y represent a file of 100 KB. If x is divided into a "chunk" of size y, the total number of chunks to be time-locked would be $\frac{x}{y}$. The total time required to time-lock file x when divided into y-size chunks would be a fraction of the time required to encrypt file x itself. This can be demonstrated as

$$(\frac{x}{y})y^2 \le x^2 \tag{25}$$

for $x = 200$, $y = 100$

$$(\frac{200}{100})100^2 \le 200^2 \tag{26}$$

$$(2)100^2 \le 40,000 \tag{27}$$

$$20,000 \leq 40,000 \tag{28}$$

A time-lock puzzle creation that requires exponential time to be created, when divided, can be expedited relative to the original size of the file. The time required to lock a file larger than 100 KB is reduced to a multiple of the 100 KB encryption time. The efficiency of this "chunked" encryption approach increases with increasing file size of content to be time-locked.

This "chunked" time-lock data model has been included in `mod_oai` to benefit from this observed speedup during during dynamic data dissemination. A chunk size of 10 KB has been selected for implementation in `mod_oai` due to its property of requiring about to 0.2 CPU seconds for time-lock computation. Any file residing in the website with file size greater than 10 KB that is to be included as an encrypted datastream in the resulting XML document has been divided into 10 KB chunks to achieve faster encryption time.

The website harvest performance test has been repeated using chunked time-lock encryption in `mod_oai` with the same *modoai_encode_size* values as displayed in Table VI. The following table records the amount of time required to harvest the entire website using chunked time-lock encryption with increasing *modoai_encode_size* and by-value content inclusion. The table below records an average of five harvest time values along with the standard deviation and speedup achieved in comparison with regular time-locked encoding. An increase in *modoai_encode_size* allows records with large file size to be included in the XML response as encoded bytestream. Chunked time-lock encoding increases in efficiency with increase in file size, leading to a significant overall harvest time speedup.

The initial system evaluation harvest times recorded in Table VI have been combined with the harvest times taken during chunked data encryption, as shown in the Table VII, for a graphical comparison in speedup. The harvest times and *modoai_encode_size* have been plotted in log scale to correctly display and compare the wide range of harvest time values.

A comparison of the plotted times between regular time-lock encryption and chunked encryption reveals a 70× speedup in harvest times. Even though an exponential increase in harvest time is still observed, the chunked harvest time curve has been "pushed" to the "right" for common file sizes, resulting in a slower increase in exponential harvest time. As observed from the graph, with *modoai_encode_size* set to 15

modoai_encode_size (Bytes)	No. of Responses	Chunked Encryption (sec)	σ (sec)	Speedup
150,000	70	99	7.7	6
300,000	38	101	12.4	6
500,000	24	102	7.9	9
700,000	16	103	6.5	10
1,000,000	13	108	10.7	9
5,000,000	6	161	16.1	23
10,000,000	6	160	16.2	65
15,000,000	6	156	16.9	70

TABLE VII: Wallclock harvest times of website with using "chunked" time-lock encryption.

MB, chunked encryption results in a speedup of 70, with the harvest time reduced from 3 hours and 2 minutes to only 2.6 minutes. This time penalty for disseminating embargoed content is within the realistic, feasible range of website harvest time. This harvest time is proportional to the total size of the website being harvested, as well as the average size of files that are encrypted.

VII.2   MPEG21 DIDL DOCUMENT FORMAT MODIFICATIONS

The MPEG21 DIDL document format represented in Figure 5 has been modified to reflect the inclusion of chunked time-lock encryption. In this optimization, the size of each encrypted chunk is set to 10 KB. Every file whose file size is greater than 10 KB has been divided into 10 KB chunks and individually encrypted. Each file chunk has been encapsulated into one *component*. Therefore, each chunked encrypted file, contained in a DIDL *record* entity, contains multiple *components*. Chunked *records* in the exported XML document can be identified by their file size, as well as the number of *components* contained in each *record*. Figure 11 is a graphical representation of this DIDL document model. The items in bold are modifications or additions from the original model utilized during data harvest without embargoed data encryption. A corresponding XML document to a GetRecord request has been included in appendix B.2.
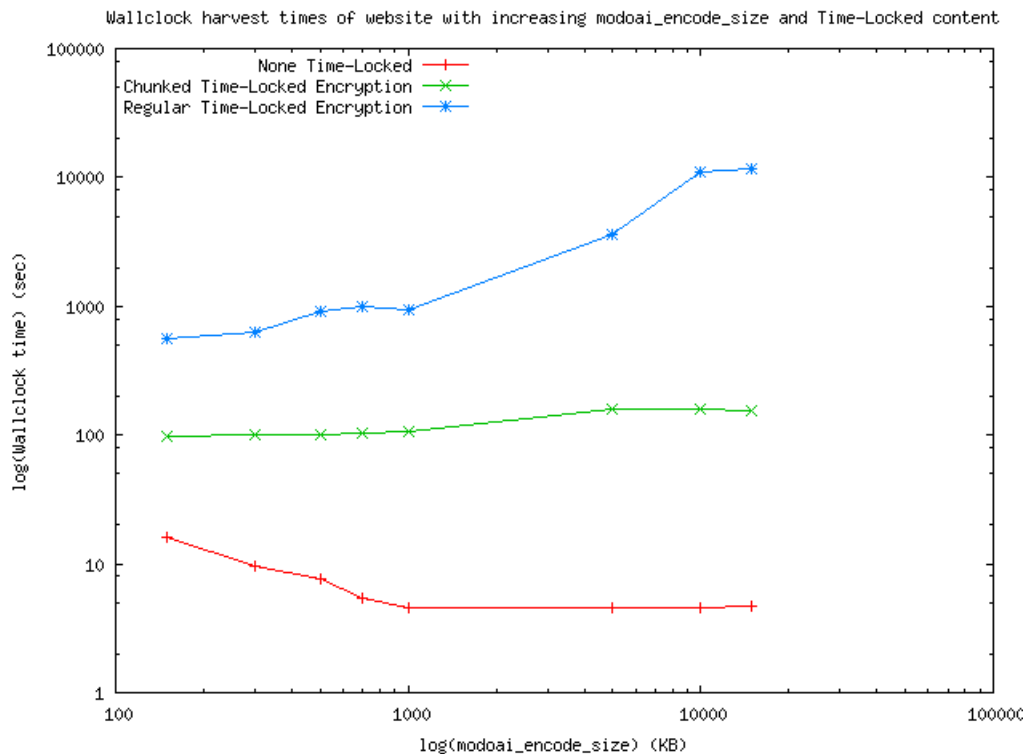
FIG. 10: Harvest times of website using no data time-lock, regular time-lock and chunked time-lock encryption during dynamic data dissemination.

Each record *component* needs to be identified for accurate reordering, decryption and reassembly of contained data chunks into one file. These *components* have been associated with Identifiers in increasing lexicographical order, which have been encapsulated in an *identifier* entity within each component. A *descriptor* has also been inserted in each *component* to provide additional information regarding the total number of chunks contained in the record to ensure that each isolated component contains sufficient information required for reassembly of chunks into one file. This modified document model still contains the original record identifier and related metadata for record identification.

VII.3 SUMMARY

This chapter introduced the concept of chunked time-lock encryption, division of one file into set file-size chunks for faster time-lock puzzle creation. It evaluates this

chunked encryption model with regular time-lock encryption to conclude a speedup of 70, making this data harvest model reliable and feasible.
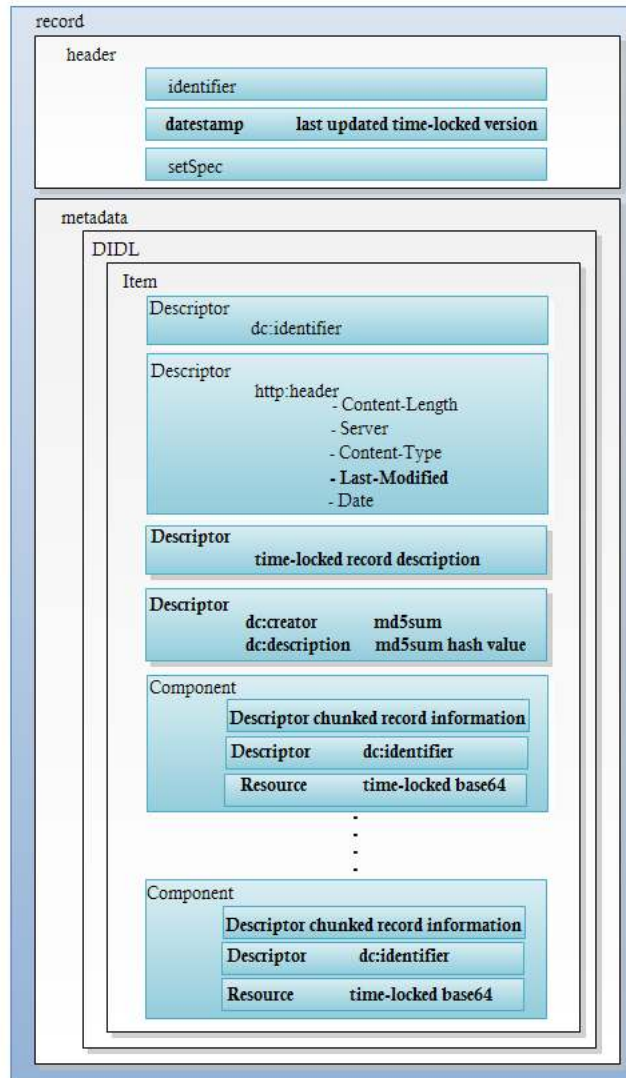
FIG. 11: MPEG-21 DIDL Document format of a record time-locked using chunked encryption.

CHAPTER VIII

FUTURE CONSIDERATIONS

In keeping with its aims, this thesis research has developed the Time-Locked Embargo framework to successfully disseminate time-locked instances of embargoed content via `mod_oai` data harvesting module. Several issues addressed in this framework can be refined with future research.

VIII.1   CHUNK SIZE PERFORMANCE DEPENDENCY

Further research may be conducted to ensure that the variables that may affect the performance and harvest time of the `mod_oai` harvesting module have been modified according to the environment and computer system being utilized. As discussed in Chapter VII, this research has optimized the utilization of time-lock puzzle creation via chunked encryption. The default chunk size in this optimization has been set to 10 KB, as this was the closest file size that could be time-locked in 0.2 CPU seconds during performance testing. It is suggested that further tests of time-lock puzzle creation without chunked encoding may be conducted on the computer systems being utilized during framework installation to determine the optimal chunk size. This chunk size may vary according to the machine configuration being used.

VIII.2   OTHER OPTIMIZATION METHODS

Due to the exponential time required to time-lock a file, this research has adopted the chunked encryption optimization model. Various other optimization approaches can be further researched and implemented in `mod_oai` to explore the most robust model.

VIII.2.1   PARALLELISM

Files residing in the website to be harvested can be time-locked in parallel in order to decrease the amount of time spent in data encryption and total data harvest time. This optimization model would require more dedicated resources at the time of a data harvesting request. Further research may be required to find an optimum balance between the utilization of resources versus time.

VIII.2.2   DATA PRELOCKING

A machine may be dedicated to time-lock embargoed data at every decrement in remaining embargo period. This new instance of decreased time-lock record can be saved on the machine and utilized during data dissemination to reduce the time overhead of dynamic time-lock puzzle creation for each embargoed file residing in the website. This model of prelocking embargoed content would result in faster data harvesting upon an OAI-PMH data request.

VIII.2.3   HYBRID APPROACH: TIME-LOCKING THE ENCRYP-TION KEY

The time overhead required to create time-lock puzzles of embargoed content can also be reduced by adopting a hybrid approach to content encryption. The embargoed content may be encrypted using a standard, widely recognized and adopted encryption method. A key k is generated, which can be further utilized for embargoed data encryption. This key can be time-locked for the embargo period of the embargoed content itself. Presuming that the key is required for successful decryption and access of the embargoed record, the amount of time required to break and access the encryption key is equivalent to the embargo period of the file, ensuring that the key cannot be accessed until the embargo period of the content has elapsed.

This hybrid approach compromises the security of the embargoed data being disseminated because the security of the embargoed content itself relies on the credibility of the encryption standard being used. Since the embargoed content is not time-locked, any method discovered to break the encryption standard would breach the security of the embargoed content itself, and the effort to time-lock the encryption key would be exhausted.

This hybrid approach would reduce, and almost eliminate, the time required to time-lock large files, but it introduces a security risk associated with the data being disseminated. This optimization model may be adopted to mitigate the harvest time overhead with an associated security risk to embargoed content.

This research has been aimed at the development of a robust yet secure, independent system. Although further research may be required to determine the optimal implementation of a time-lock puzzle creation model, the implemented chunked encryption model has been incorporated in this thesis research due to its utilization of

minimal resources and the security of the dissemination hazard.

CHAPTER IX

CONCLUSION

Preservation of digital content is one of the foremost concerns of the scholarly community. Successful preservation of embargoed content requires not only preservation of bits, but also preservation of data access and security. This requirement restricts the use of data refreshing as a digital preservation method, as embargoed content can be distributed only to subscribed users. This thesis research has incorporated timed-release encryption, via time-lock puzzle creation, into the `mod_oai` metadata harvesting module to facilitate data harvest of time-locked instances of embargoed data.

The thesis research has introduced the "Preservation Risk Interval" problem associated with embargoed content caused due to limited diffusion of embargoed scholarly material within the digital library community. The Time-Locked Embargo framework for the mitigation of this risk has been recommended in Chapter V that introduces timed-release encryption of embargoed content during data dissemination. The framework introduces the identification of embargoed content and calculates the required complexity of the time-lock puzzle to be created for that content. During the integration of time-lock puzzles into `mod_oai`, an initial system evaluation revealed that the amount of time required to create time-lock puzzles during dynamic DIDL document creation increases exponentially with the size of the embargoed file. This exponential increase in time overhead has been reduced by incorporating "chunked" time-lock encryption: the division of files into set-size, 10 KB data chunks for faster encryption and data harvest time. The framework has also modified the MPEG-21 DIDL complex object format utilized by `mod_oai` to accurately encapsulate chunked embargoed content and related metadata. This amalgam of concepts has resulted in a prototype of `mod_oai` that is successfully able to disseminate time-locked datastreams of embargoed content encapsulated in a DIDL document upon an OAI-PMH data harvest request.

Chapter VI described various experiments conducted to ensure that embargoed content can be successfully time-locked for a pre-determined amount of time and cannot be broken till the desired amount of computation time, equivalent to the embargo time period of the content, has passed. Chapter VII presented an evaluation of the optimized chunked encryption system, whereby demonstrating that embargoed

content can be time-locked during data dissemination with an acceptable and feasible time overhead.

This evaluation of the system demonstrates that this thesis research has fulfilled the aim of developing the Time-Locked Embargo framework that mitigates the Preservation Risk Interval associated with embargoed content. With the use of the expanded `mod_oai` module, resources under embargo can be exchanged between a much broader scholarly community for the purpose of digital preservation as well as content diffusion.

BIBLIOGRAPHY

[1] ISO 14721:2003. Reference Model for an Open Archival Information System (OAIS). CCSDS 650.0-B-1. Blue Book, January 2002.

[2] William Y. Arms. Preservation of scientific serials: Three current examples. *Journal of Electronic Publishing*, 5(2), December 1999.

[3] William Y. Arms. *Digital libraries*. MIT Press, Cambridge, Ma., 2000.

[4] Mary Baker, Mehul Shah, David S. H. Rosenthal, Mema Roussopoulos, Petros Maniatis, TJ Giuli, and Prashanth Bungale. A Fresh Look at the Reliability of Long-term Digital Storage. *SIGOPS Oper. Syst. Rev.*, 40(4):221–234, 2006.

[5] Vannevar Bush. As we may think. *The Atlantic Monthly*, pages 101–108, July 1945.

[6] Leonardo Candela, Fuat Akal, Henri Avancini, Donatella Castelli, Luigi Fusco, Veronica Guidetti, Christoph Langguth, Andrea Manzi, Pasquale Pagano, Heiko Schuldt, Manuele Simi, Michael Springmann, and Laura Voicu. DILIGENT: integrating digital library and Grid technologies for a new Earth observation research infrastructure. *Int. J. Digit. Libr.*, 7(1):59–80, 2007.

[7] Donatella Castelli. DILIGENT: A Digital Library Infrastructure on Grid Enabled Technology. *ERCIM News*, (59):26–27, October 2004.

[8] Sherman S.M. Chow, Volker Roth, and Eleanor G. Rieffel. General Certificateless Encryption and Timed-Release Encryption. Cryptology ePrint Archive, Report 2008/023, 2008. http://eprint.iacr.org/.

[9] Giovanni Di Crescenzo, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional Oblivious Transfer and Timed-Release Encryption. *Lecture Notes in Computer Science*, 1592:74–89, 1999.

[10] David H. Crocker. RFC822 - Standard for the Format of ARPA Internet Text Messages, 1982.

[11] Philip M. Davis. Do Open-Access articles really have a greater research impact? *College & Research Libraries*, 67(2), 2006.

[12] Herbert Van de Sompel and Carl Lagoze. The Santa Fe Convention of the Open Archives Initiative. *D-Lib Magazine*, 6(2), 2000.

[13] Eileen Gifford Fenton. An overview of portico: An electronic archiving service. *Serials Review*, 32(2):81–86, June 2006.

[14] Miguel Ferreira, Ana Alice Baptista, and José Carlos Ramalho. A foundation for automatic digital preservation. *Ariadne*, 48, July 2006.

[15] Miguel Ferreira, Ana Alice Baptista, and José Carlos Ramalho. An intelligent decision support system for digital preservation. *Int. J. Digit. Libr.*, 6(4):295–304, 2007.

[16] Juan A. Garay and Markus Jakobsson. Timed release of standard digital signatures. In *Financial Cryptography 02*, pages 168–182. Springer-Verlag, 2002.

[17] Stevan Harnad, Tim Brody, Francois Vallieres, Les Carr, Steve Hitchcock, Yves Gingras, Charles Oppenheim, Heinrich Stamerjohanns, and Eberhard R. Hilf. The access/impact problem and the green and gold roads to open access. *Serials Review*, 30(4):310–314, 2004.

[18] Steven Harnad and Tim Brody. Comparing the Impact of Open Access (OA) vs. Non-OA Articles in the Same Journals. *D-Lib Magazine*, 10(6), 2004.

[19] Sam Harrell, Tom Seidel, and Bernard Fay. The national technology roadmap for semiconductors and sematech future directions. *Microelectron. Eng.*, 30(1-4):11–15, 1996.

[20] Stephen P. Harter. Scholarly Communication and the Digital Library: Problems and Issues. *Journal of Digital Information*, 1(1), 1997.

[21] Thomas B. Hickey Herbert Van de Sompel, Jeffery A. Young. Using the OAI-PMH Differently. *D-Lib Magazine*, 9(7/8), July/August 2003.

[22] J. R. Van Der Hoeven, R. J. Van Diessen, and K. Van Der Meer. Development of a Universal Virtual Computer (UVC) for long-term preservation of digital objects. *Journal of Information Science*, 31(3):196–208, 2005.

[23] D. Hristu-Varsakelis, K. Chalkias, and G. Stephanides. Low-cost anonymous timed-release encryption. In *IAS '07: Proceedings of the Third International*

Symposium on Information Assurance and Security, pages 77–82, Washington, DC, USA, 2007. IEEE Computer Society.

[24] J. Hunter and S. Choudhury. Semi-automated preservation and archival of scientific data using semantic grid services. In *CCGRID '05: Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05) - Volume 1*, pages 160–167, Washington, DC, USA, 2005. IEEE Computer Society.

[25] Jane Hunter and Sharmin Choudhury. Implementing preservation strategies for complex multimedia objects. In *Proc. 7th European Conf. Research and Advanced Technology for Digital Libraries, ECDL 2003, Aug 17-22*, pages 473–486, Trodheim, Norway, 2003.

[26] Jane Hunter and Sharmin Choudhury. A semi-automated digital preservation system based on semantic web services. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 269–278, New York, NY, USA, 2004. ACM.

[27] Jane Hunter and Sharmin Choudhury. PANIC: An integrated approach to the preservation of composite digital objects using Semantic Web Services. *Int. J. Digit. Libr.*, 6(2):174–183, 2006.

[28] ISO/IEC. ISO/IEC 21000-2:2005 Information Technology - Multimedia Framework (MPEG-21) - Part 2: Digital Item Declaration - Schema for derived DIDL types.

[29] Herbert Van de Sompel Jeroen Bekaert, Emiel De Kooning. Representing Digital Assets using MPEG-21 Digital Item Declaration. *International Journal on Digital Libraries*, 6(2):159–173, 2006.

[30] J. Jonsson and B. Kaliski. Public-Key Cryptography Standards (PKCS) 1: RSA Cryptography Specifications Version 2.1, 2003. RFC 3447.

[31] Charles W. Bailey Jr. The Spectrum of E-Journal Access Policies: Open to Restricted Access. http://dlist.sir.arizona.edu/990/01/spectrum.htm, 2005.

[32] Robert Kahn and Robert Wilensky. A framework for distributed digital object services. Technical Report cnri.dlib/tn95-1, CNRI, Reston, VA, 1995.

[33] G. Klyne and C. Newman. RFC3339 - Date and Time on the Internet: Timestamps, 2002.

[34] Paul Koerbin. The PANDORA Digital Archiving System (PANDAS): Managing Web Archiving in Australia: A Case Study. Bath, UK, 16th September 2004.

[35] Carl Lagoze and Herbert Van de Sompel. The Open Archives Initiative: building a low-barrier interoperability framework. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 54–62, New York, NY, USA, 2001. ACM Press.

[36] Steve Lawrence. Online or Invisible? *Nature*, 411(6837):521, 2001.

[37] Michael Lesk. *Practical Digital Libraries: Books, Bytes, and Bucks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[38] Michael Lesk. *Understanding Digital Libraries*. Morgan Kaufmann, 2nd edition, December 2004.

[39] Raymond A. Lorie. A methodology and system for preserving digital data. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 312–319, New York, NY, USA, 2002. ACM.

[40] Clifford A. Lynch. Metadata harvesting and the open archives initiative. Technical Report 217, ARL: A Bimonthly Report, August 2001.

[41] P. Maniatis, M. Roussopoulos, T. Giuli, D. Rosenthal, M. Baker, and Y. Muliadi. Preserving peer replicas by rate-limited sampled voting. Technical Report arXiv:cs.CR/0303026, Stanford University, Stanford, March 2003.

[42] Wenbo Mao. Timed-release cryptography. *Lecture Notes in Computer Science*, 2259:342–357, 2001.

[43] Nancy Y. McGovern, Anne R. Kenney, Richard Entlich, William R. Kehoe, and Ellie Buckley. Virtual Remote Control. Building a Preservation Risk Management Toolbox for Web Resources. *D-Lib Magazine*, 10(4), 2004.

[44] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117, 1965.

76

[45] Terry Morrow, Neil Beagrie, Maggie Jones, and Julia Chruszcz. A Comparative study of e-Journal Archiving Solutions. Technical report, JISC, 2008.

[46] Michael L. Nelson, Joan A. Smith, Ignacio, Herbert Van de Sompel, and Xiaoming Liu. Efficient, automatic web resource harvesting. In *WIDM '06: Proceedings of the eighth ACM international workshop on Web information and data management*, pages 43–50, New York, NY, USA, 2006. ACM Press.

[47] Michael L. Nelson, Herbert Van de Sompel, Xiaoming Liu, Terry L. Harrison, and Nathan McFarland. mod_oai: An Apache module for metadata harvesting. In *Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005)*, pages 509–510, 2005.

[48] R.J. van Diessen N.J.C. Kol and K. van der Meer. An improved Universal Virtual Computer approach for long-term preservation of digital objects. *Information Services and Use*, 26(4):283–291, 2006.

[49] OAI-PMH. History and Development of OAI-PMH. http://www.oaforum.org/tutorial/english/page2.htm, 2003.

[50] A. Odlyzko. Competition and cooperation: Libraries and publishers in the transition to electronic scholarly journals. *J. Scholarly Publishing*, 30(4):163–185, 1999.

[51] Bangalore: Indian Institute of Science. Bangalore declaration: A national open access policy for developing countries. http://www.ncsi.iisc.ernet.in/OAworkshop2006/pdfs/NationalOAPolicyDCs.pdf, 2006.

[52] Erik Oltmans and Adriaan Lemmen. The e-Depot at the National Library of the Netherlands. *Serials: The Journal for the Serials Community*, 19(1):61–67, March 2006.

[53] Erik Oltmans, Raymond van Diessen, and Hilde van Wijngaarden. Preservation functionality in a digital archive. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 279–286, New York, NY, USA, 2004. ACM.

[54] Budapest Open Access Initiative. Budapest Open Access Initiative. http://www.soros.org/openaccess/read.shtml, 2007.

[55] Budapest Open Access Initiative. Important Open Access Initiatives. http://www.soros.org/openaccess/initiatives.shtml, 2007.

[56] Margaret Phillips. PANDORA, Australia's Web Archive, and the Digital Archiving System that Supports it, December 2003.

[57] J. Pringle. Do open access journals have impact? http://www.nature.com/focus/accessdebate/19.html, 2004.

[58] G. Drury R. Van de Walle, I. Burnett. ISO/IEC 21000-2 Digital Item Declaration (Output Document of the 70th MPEG Meeting, Palma De Mallorca, Spain, No. ISO/IEC JTC1/SC29/WG11/N6770)., October 2004. Retrieved from the NIST MPEG Document Register.

[59] Vicky Reich and David S.H. Rosenthal. LOCKSS: A permanent web publishing and access system. *D-Lib Magazine*, 7(6), 2001.

[60] Victoria A. Reich. Diffused knowledge immortalizes itself, the LOCKSS program. High Energy Physics Libraries Webzine, October 2002.

[61] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[62] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical Report 684, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.

[63] Ronald L. Rivest. The MD5 Message-Digest Algorithm (RFC 1321). http://www.ietf.org/rfc/rfc1321.txt?number=1321.

[64] Ronald L. Rivest. Description of the LCS35 Time Capsule Crypto-Puzzle, April 1999.

[65] RLG. Preserving digital information: Report of the task force on archiving of digital information. http://www.rlg.org/ArchTF/, 1996.

[66] A. Robinson. Open access: the view of a commercial publisher. *Journal of Thrombosis and Haemostatis*, 4(7):1454–1460, 2006.

[67] David S. Rosenthal, Thomas Robertson, Tom Lipkis, Vicky Reich, and Seth Morabito. Requirements for Digital Preservation Systems: A Bottom-Up Approach. *D-Lib Magazine*, 11(11), November 2005.

[68] David S. H. Rosenthal, Thomas Lipkis, Thomas Robertson, and Seth Morabito. Transparent format migration of preserved web content. *D-Lib Magazine*, 11(1), January 2005.

[69] Michael Seadle. A social model for archiving digital serials: Lockss. *Serials Review*, 32(2):73–77, 2006.

[70] SHERPA. Definitions and terms. http://www.sherpa.ac.uk/romeoinfo.html, 2006.

[71] SHERPA. Journal policies - summary statistics so far. http://romeo.eprints.org/stats.php, 2007.

[72] Peter Suber. Timeline of the Open Access Movement. http://www.earlham.edu/p̃eters/fos/timeline.htm, February 2007.

[73] Hussein Suleman and Edward Fox. The open archives initiative: Realizing simple and effective digital library interoperability. *Journal of Library Administration*, 35:125–146, 2001.

[74] A. Swan and S. Brown. Authors and open access publishing. *Learned Publishing*, 7(3):219–224, 2004.

[75] Alma Swan and Sheridan Brown. JISC/OSI Journal Authors Survey Report, 2004.

[76] Herman te Riele. Factoring large numbers. *ERCIM News*, 1995. SPECIAL: Computational Mathematics.

[77] Jim Tucker and Mary Sue Hoyle. Understanding embargoes and utilizing other services. *The Serials Librarian*, 45(3):115–117, 2003.

[78] Herbert Van de Sompel and Carl Lagoze. Notes from the interoperability front: A progress report on the Open Archives Initiative. In *ECDL '02: Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 144–157, London, UK, 2002. Springer-Verlag.

[79] Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner. Resource harvesting within the OAI-PMH framework. *D-Lib Magazine*, 10(12), 2004.

[80] Hilde van Wijngaarden. Long-term preservation and permanent access: How to ensure the long-term reuse value of your digital assets. *Journal of Digital Asset Management*, 3(2):102–109, April 2007.

[81] H. G. Wells. *World Brain.* Meuthuen Co. Limited, 1938.

[82] Ian H. Witten and David Bainbridge. *How to Build a Digital Library.* Elsevier Science Inc., New York, NY, USA, 2002.

APPENDIX A

DYNAMIC EMBARGOED RECORD IDENTIFICATION

A.1 VARIABLES UTILIZED DURING RECORD IDENTIFICATION

Following is the `mod_oai` Apache module configuration file containing the input variables utilized during the formulation of a record time-lock puzzle. Four new variables have been included in the configuration file that are required for the implementation of embargoed record identification and encapsulation.

< Location/modoai >	
SetHandler	modoai-handler
modoai_admin	admin
modoai_email	admin@modoai
modoai_oai_active	ON
modoai_encode_size	5000000
modoai_resumption_count	100
lock_start	2008-01-01T12:00:00Z
duration	365
interval	12
modoai_encode_size	10000
< /Location >	

$lock\_start$, $duration$ and $interval$ are used during Embargoed Record Identification and Encryption. $lock\_start$ is the globally defined $publisher_{start}$ datestamp. Any record that is published after this datestamp is considered to be under a publisher-imposed embargo for the number of days set by the variable $duration$. $interval$ corresponds to the $embargo_{decrement}$ variable, that sets the number of update intervals for each record under embargo. The above configuration file imposes an embargo on all records published after the datestamp 2008-01-01T12:00:00Z. The length of the embargo period is 365 days, with 12 record updates before an unlocked instance of that record is published. $modoai\_encode\_size$, defined in bytes, is used during Embargoed Record Encapsulation. This variable sets the chunk size of the record during chunked encoding. A $modoai\_encode\_size$ of 10000 bytes means that any file that has the file size of 10000 or greater would be time-locked using chunked encoding, with chunks of a maximum size of 10000 bytes.

APPENDIX B

EMBARGOED RECORD OAI-PMH RESPONSE

B.1   EMBARGOED RECORD *GETRECORD* RESPONSE

Following is the DIDL document response to an OAI-PMH GetRecord request of a
record under embargo. It reflects the structural changes highlighted in figure 5.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2008-09-30T18:15:50Z</responseDate>
<request verb="GetRecord" identifier="http://isis.cs.odu.edu:10321/code/ExtraDescriptor.txt"
 metadataPrefix="oai_didl">http://isis.cs.odu.edu:10321/modoai/</request>
<GetRecord>
<record>
<header>
<identifier>http://isis.cs.odu.edu:10321/code/ExtraDescriptor.txt</identifier>
<datestamp>2008-09-12T16:38:16Z</datestamp>
<setSpec>mime:text:plain</setSpec>
</header>
<metadata>
<didl:DIDL xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS http://purl.lanl.gov/STB-RL/schemas/2004-11/DIDL.xsd">
<didl:Item>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">http://isis.cs.odu.edu:10321/code/ExtraDescriptor.txt</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<http:header xmlns:http="http://www.modoai.org/OAI/2.0/http_header/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.modoai.org/OAI/2.0/http_header/ http://purl.lanl.gov/STB-RL/schemas/2004-08/HTTP-HEADER.xsd">
<http:Content-Length>774</http:Content-Length>
<http:Server>Apache/2.0.54 (Unix)</http:Server>
<http:Content-Type>text/plain</http:Content-Type>
<http:Last-Modified>Fri, 12 Sep 2008 16:38:16 GMT</http:Last-Modified>
<http:Date>Tue, 30 Sep 2008 18:15:50 GMT</http:Date>
</http:header>
</didl:Statement>
</didl:Descriptor>
<didl:Component>
<didl:Resource mimeType="text/plain" encoding="base64">MzEzMjM5MTc0ONT ... DcyMTIzMjg3MDUzOQ==</didl:Resource>
<didl:Resource mimeType="text/plain"
 ref="http://isis.cs.odu.edu:9321/timelock.cgi?uri=http://isis.cs.odu.edu:10321/code/ExtraDescriptor.txt"/>
</didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<dc:creator xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://purl.org/dc/elements/1.1/ http://dublincore.org/schemas/xmls/simpledc20021212.xsd">
md5sum (GNU coreutils) 6.9
Copyright (C) 2007 Free Software Foundation, Inc.
This is free software.  You may redistribute copies of it under the terms of
the GNU General Public License http://www.gnu.org/licenses/gpl.html.
There is NO WARRANTY, to the extent permitted by law.

Written by Ulrich Drepper, Scott Miller, and David Madore.
</dc:creator>
<dc:description xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://purl.org/dc/elements/1.1/
```

```
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">9d53e9ff7f38bc446ae7edc91f9e74b5
 </dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<dc:description xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://purl.org/dc/elements/1.1/ http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been
 time-locked since 2008-02-12T17:38:16Z; see MIT/LCS/TR-684 (February 1996) for more information. This version of the record is 7 of
 12 separate encryptions, each of which is successively easier to break. It will take approximately 3650 hours of computation to break this
 time-lock.  The next update will be available on 2008-10-13T02:38:16Z. Puzzle Parameters (all in decimal): n = 398399 t = 73602000000.
 To solve the puzzle, first compute w = 2^(2^t) (mod n). Then exclusive-or the result with the resource. The result is the secret message (8
 bits per character).
</dc:description>
</didl:Statement>
</didl:Descriptor>
</didl:Item>
</didl:DIDL>
</metadata>
</record>
</GetRecord>
</OAI-PMH>
```

B.2 EMBARGOED RECORD *GETRECORD* RESPONSE USING CHUNKED ENCODING

Following is the XML response to a GetRecord request of a record that has been time-locked using chunked encoding. The filesize of the record is 63632 bytes, which is greater than the set *modoai_oai_encode* size of 10000 bytes, and thus has been time-locked in chunks of 10000 bytes. It reflects the structural changes outlined in figure 11.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2008-09-30T18:04:19Z</responseDate>
<request verb="GetRecord" identifier="http://isis.cs.odu.edu:10321/oduthesis/datestamp_outline.JPG"
 metadataPrefix="oai_didl">http://isis.cs.odu.edu:10321/modoai/</request>
<GetRecord>
<record>
<header>
<identifier>http://isis.cs.odu.edu:10321/oduthesis/datestamp_outline.JPG</identifier>
<datestamp>2008-09-25T17:55:04Z</datestamp>
<setSpec>mime:image:jpeg</setSpec>
</header>
<metadata>
<didl:DIDL xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS http://purl.lanl.gov/STB-RL/schemas/2004-11/DIDL.xsd">
<didl:Item>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">http://isis.cs.odu.edu:10321/oduthesis/datestamp_outline.JPG</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<http:header xmlns:http="http://www.modoai.org/OAI/2.0/http_header/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.modoai.org/OAI/2.0/http_header/ http://purl.lanl.gov/STB-RL/schemas/2004-08/HTTP-HEADER.xsd">
<http:Content-Length>63632</http:Content-Length>
```

```
<http:Server>Apache/2.0.54 (Unix)</http:Server>
<http:Content-Type>image/jpeg</http:Content-Type>
<http:Last-Modified>Thu, 25 Sep 2008 17:55:04 GMT</http:Last-Modified>
<http:Date>Tue, 30 Sep 2008 18:04:19 GMT</http:Date>
</http:header>
</didl:Statement>
</didl:Descriptor>
<didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">
AAAAAA</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dc:description xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been split into 10000-byte chunks for faster processing.
This is part 1 of 7 chunks, with unlocked chunks to be reassembled in the specified order.</dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Resource mimeType="image/jpeg" encoding="base64">NDIyNzM5NDDcwND ... Y2MjQwMjA4MDkxMTMw</didl:Resource>
</didl:Component>
<didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">
AAAAAB</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dc:description xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been split into 10000-byte chunks for faster processing.
 This is part 2 of 7 chunks, with unlocked chunks to be reassembled in the specified order.</dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Resource mimeType="image/jpeg" encoding="base64">NDIxMDk0NTk4NTgx ... TgxNjE3NzgwNDI4NDMw</didl:Resource>
</didl:Component>
<didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">
AAAAAC</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dc:description xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been split into 10000-byte chunks for faster processing.
 This is part 3 of 7 chunks, with unlocked chunks to be reassembled in the specified order.</dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Resource mimeType="image/jpeg" encoding="base64">NDIxMDk0NTY1MDg5O ... TI3OTQzODQ4Mjc3MzA4</didl:Resource>
</didl:Component>
<didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">
AAAAAD</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dc:description xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been split into 10000-byte chunks for faster processing.
```

This is part 4 of 7 chunks, with unlocked chunks to be reassembled in the specified order.</dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Resource mimeType="image/jpeg" encoding="base64">NjQ1NDMzMjExOTQx ... ODM4NjkyMDYxMjk3Nw==</didl:Resource>
</didl:Component>
<didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">
AAAAAE</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dc:description xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been split into 10000-byte chunks for faster processing.
 This is part 5 of 7 chunks, with unlocked chunks to be reassembled in the specified order.</dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Resource mimeType="image/jpeg" encoding="base64">NDIyNzMzMDQxMTY1 ... MjY0NzQ1ODgwMDc3</didl:Resource>
</didl:Component>
<didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">
AAAAAF</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dc:description xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been split into 10000-byte chunks for faster processing.
This is part 6 of 7 chunks, with unlocked chunks to be reassembled in the specified order.</dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Resource mimeType="image/jpeg" encoding="base64">NDIyNzMzMDMMOMT ... U5MTM3MDgwMjIyNjg5</didl:Resource>
</didl:Component>
<didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dii:Identifier xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mpeg:mpeg21:2002:01-DII-NS
 http://purl.lanl.gov/STB-RL/schemas/2003-09/DII.xsd">
AAAAAG</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="image/jpeg"><dc:description xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/
 http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been split into 10000-byte chunks for faster processing.
 This is part 7 of 7 chunks, with unlocked chunks to be reassembled in the specified order.</dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Resource mimeType="image/jpeg" encoding="base64">MjMONjEzNjNjIxMzcxMDA5 ... zAyMjI3NTA2MzYwMTc=</didl:Resource>
</didl:Component>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<dc:creator xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://purl.org/dc/elements/1.1/ http://dublincore.org/schemas/xmls/simpledc20021212.xsd">
md5sum (GNU coreutils) 6.9
Copyright (C) 2007 Free Software Foundation, Inc.
This is free software. You may redistribute copies of it under the terms of
the GNU General Public License http://www.gnu.org/licenses/gpl.html.
There is NO WARRANTY, to the extent permitted by law.

Written by Ulrich Drepper, Scott Miller, and David Madore.
</dc:creator>
<dc:description xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://purl.org/dc/elements/1.1/

```
  http://dublincore.org/schemas/xmls/simpledc20021212.xsd">465d18c06834892fcc69d7e9438a29dd
 </dc:description>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor>
<didl:Statement mimeType="application/xml; charset=utf-8">
<dc:description xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://purl.org/dc/elements/1.1/ http://dublincore.org/schemas/xmls/simpledc20021212.xsd">This record has been
 time-locked since 2008-08-26T06:55:04Z; see MIT/LCS/TR-684 (February 1996) for more information. This version of the record is 1 of
 12 separate encryptions, each of which is successively easier to break. It will take approximately 8030 hours of computation to break this
 time-lock.  The next update will be available on 2008-10-26T03:55:04Z. Puzzle Parameters (all in decimal): n = 398399 t =
 161929800000. To solve the puzzle, first compute w = 2^(2^t) (mod n). Then exclusive-or the result with the resource. The result is the
 secret message (8 bits per character).
</dc:description>
</didl:Statement>
</didl:Descriptor>
</didl:Item>
</didl:DIDL>
</metadata>
</record>
</GetRecord>
</OAI-PMH>
```

VITA

Rabia Haq
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

EDUCATION
M.S. in Computer Science, Old Dominion University, 2008
B.S. in Computer Science,Old Dominion University, 2004

EMPLOYMENT
03/07 to Present Software Engineer at C& F Enterprises, Inc.
01/06 to 03/06   Software Quality Assurance Intern at Symantec, Inc.
09/04 to 12/05   Research and Teaching Assistant at Old Dominion University

PROFESSIONAL AFFILIATIONS
Association for Computing Machinery (ACM)
Golden Key International Honor Society

Typeset using LATEX.