Maik Schott[a], Jana Dittmann[a], Claus Vielhauer[a,b], Christian Krätzer[a], Andreas Lang[a]

a) Otto-von-Guericke University Magdeburg
{mschott|dittmann|vielhaue|kraetzer|alang}@iti.cs.uni-magdeburg.de

b) University of Applied Sciences in Brandenburg
claus.vielhauer@fh-brandenburg.de

INTEGRITY AND AUTHENTICITY FOR DIGITAL LONG-TERM PRESERVATION IN IRODS GRID INFRASTRUCTURE

Abstract: Digital resources and documents have become part of our culture as many cultural and intellectual goods are "born-digital" – existing only in digital form. Therefore digital long-term preservation is sustaining our cultural heritage for the future. Because of its importance digital preservation is addressed by several national and international projects like the German NESTOR project [12], the US National Digital Information Infrastructure and Preservation Program [11], the IETF LTANS group ([21], [22]) and the EU FP7 SHAMAN Integrated Project [18].

Preserved resources and goods are only useful if their integrity and authenticity can be proven at any time. Although usually not made with the explicit intention to ensure integrity and authenticity in mind, in most cases both can be proven nevertheless for non-digital entities by their inherent characteristics. By their very nature, this is not the case for digital entities. Digital preservation must also ensure the integrity and authenticity of the entire preservation environment. This further imposes issues regarding the objective of digital long term preservation for at least 100 years in combination with the fast-paced developments in information technology. Many means of security used today become useless or even long forgotten in 100 years.

For ensuring integrity and authenticity often electronic signatures are used and have been proposed for usage in digital preservation as in [17]. But digital signatures do not provide tamper protection or protect the integrity of data; they merely verify data integrity [3]. Although the obsolescence problem of digital signatures can be countered by a renewal of the signature, the tamper protection problem cannot be solved however. Especially in digital long-term preservation where by emulation and migration of data media breaks occur, e.g. conversion from TIFF to JPEG , digital signatures are therefore only of limited use.

Regarding all these hindrances it is not unusual in the digital preservation domain to only casually address these problems. In this paper we present a first study exemplary for the SHAMAN Integrated Project. This project will be based upon the iRODS grid infrastructure using so called rules and micro-services to establish an archive environment. We propose approaches how to enforce integrity and authenticity for digital objects, collections of these, micro-services and the entire archival infrastructure exemplarily in the environment by extending the Clark-Wilson integrity model with our previously introduced Syntactic and Semantic model.

Keywords: Digital long-term preservation, security, integrity, authenticity

1. Introduction

Based on previous work in [14] we present a first study how to enforce integrity and authenticity for digital preservation beyond electronic signatures to build a solution for data emulation and migration. In section 1 we define used terms and present an exemplary environment. Section 2 explains how to enforce integrity and authenticity, by selecting an applicable security model (2.1), a model for proving the integrity on a syntactic and semantic level (2.2) and how to apply this to objects (2.3), collections including the archive as a whole (2.4) and rules/services (2.5). Section 3 lists sample features, whose integrity may be verified. The last section 4 provides a summary and a conclusion.

1.1 Digital long-term preservation

According to the OAIS reference model [5], a digital preservation system consists of the technical and organizational processes ingest, archival storage, data management, administration, preservation planning and access. [5] defines each of these processes as:

At *ingest* the submitted data entities (Submission Information Package, SIP) are converted by the preservation system into one or more archival objects or Archival Information Packages (AIP). These archival objects also include metadata describing the SIP for retrieval and access and metadata for proper preservation, e.g. for enforcing confidentiality, integrity and authenticity.

After ingest the archival objects are stored by the *archival storage* process. This process is also responsible for actually preserving the archival objects or more specifically to prevent them from losing integrity (e.g. by refreshing them or prematurely migrate them to other media), to detect a loss of integrity or even recovering them (e.g. by error correction codes). The archival storage process also retrieves archival objects.

The *data management* provides services for accessing the descriptive metadata for accessing or searching archival objects satisfying certain conditions. Data management functions also include maintaining the referential integrity between the archival objects themselves and their metadata.

The *administration* is responsible for the proper operation of the whole system, establishing standards and policies, and preserving the integrity on a more organizational approach by procuring and installing new hardware and software.

The *preservation planning* ensures that the stored information remains accessible over a long time (e.g. by initializing a migration), observing the current state-of-art and developing or updating the preservation policies and standards.

The last process *access* provides services for the customers to describe the stored informations or archival objects, locating and delivering them as Dissemination Information Packages (DIP). A DIP may be composed of several AIPs or include AIPs in an altered data format, e.g. a JPEG version of an archived bitmap was requested, and include a proof of integrity and/or authenticity.

1.2 Integrity Requirements

Integrity in computer science and telecommunications refers to the integrity of resources. Integrity requirements describe how integrity of the system can be ensured (prevention) or it reports if the resource for example information is altered or manipulated (detection) or how it can be recovered into consistent state (recovery). Integrity is therefore the quality or condition of being whole, complete and unaltered. It also refers to the consistency, accuracy, and correctness [8]. For example integrity can also mean the condition in which data are identically maintained during any operation (such as transfer, storage or retrieval) or to describe the preservation of data for their intended use or specified operations. According to chapter B in the Nestor Catalogue of Criteria for Trusted Digital Repositories [12] integrity is measured in terms of those characteristics defined as valuable for preservation.

In respect to technical security mechanisms, the alteration of data can be detected, for example, by means of one-way hash functions, message authentication codes, digital signatures (especially content-based digital signatures), fragile digital watermarking, and robust digital watermarking [6].

Assuring the integrity in preservation systems of digital content refers to providing methods for preventing, tracking and verifying changes of archived objects as well as resources of the preservation system. This includes the secure sustainment, maintenance, and preservation of, for example, the storage media, the application systems and data. It has to be assured, that data can be accessed and interpreted unchanged, complete, and correct today and in the future. Procedures have to be provided to correctly access information stored and coded in today's valid data formats, schemes, and models, in the future. Finally, information and its components have to be displayed correctly today and in the future.

General requirements for preserving the integrity were given by Lipner as follows: a) users must only use existing programs, b) programmers must test their programs on a non-productive machine, c) installation must follow a special process, d) the special process must be controlled and audited, and e) system managers and auditors must have access to the system state and the logs [2].

1.3 Authenticity Requirements

Authenticity can be divided into two aspects: *data origin authenticity* is the proof of the data's origin, genuineness, originality, truth and realness. Data authenticity requirements can also be defined as prevention, detection and/or recovery requirements. The second aspect, *entity authenticity* is the proof that an entity, like a person or other agent, has been correctly identified as originator, sender or receiver; it can be ensured that an entity is the one it claims to be [8].

Entity authenticity usually is defined with identification and verification of identity of entities, such as person identity or agent identity.

Authentic information (or more general: resources) is not necessarily integer, as it might have been changed or damaged over time. But it is still the same from the same proved and authorized origin.

In respect to technical enforcement of authenticity, data origin authenticity can be achieved/proved with message authentication codes, digital signatures, fragile digital watermarking, and robust digital watermarking. Entities taking part in a communication can be proven by authentication protocols [6]. These protocols ensure that an entity is the one it claims to be.

Authenticity in digital preservation systems means, that the originality and the origin of the archived objects has to be maintained. This does not necessarily exclude changes of the archived objects, as the context and application dependent level of information loss can be defined a-priori, whereby methods for measuring the information loss have to be provided. This implies deciding, if the access to the archived object at a later time is restricted to the original object ingested to the preservation system without any change, or unrestricted. This depends on user and application requirements. In case, later and additional changes are allowed, they must be traceable and reversible, e.g. by audit trails, protocols, or a version management like a history [14].

Therefore, assuring the authenticity has to include either methods for prohibiting / avoiding / preventing changes or manipulations after an object is ingested in a preservation system or methods for tracking the changes.

Further, authenticity includes the verifiable assurance and proof of the author, participating entities like the consumer, administration, and producer as well as resources of the preservation systems like servers, networks, etc.

Other security aspects, but omitted in this first study, are availability of resources, confidentiality of informations and non-repudation. For these and a more detailed treatment of integrity and authenticity see [2].

1.4 The SHAMAN environment

The SHAMAN (Sustaining Heritage Access through Multivalent ArchiviNg) Integrated Project [18], funded by the EU 7th Framework Programme, aims to develop a preservation framework for documents, media, CAD and scientific data. SHAMAN will use the iRODS [7] grid as underlying storage system. iRODS (i Rule Oriented Data Systems) from the San Diego Supercomputer Center (SDSC) applies a rule-based and micro-service approach. To manage the data, rules are defined which come into effect immediately or delayed after a certain condition becomes true. If a rule is invoked it executes a workflow chain, which is a set of rules and micro-services (user-defined functions), and in case an error occurred a recovery chain to preserve integrity.

Another part of the SHAMAN environment will be the Multivalent engine. With this engine, files of superseded file formats can be accessed and changed even in ways not supported by the actual file format. This is done with a Virtual Machine approach without emulation, migration to a newer file format or altering the original bitstream. Therefore the integrity and authenticity of the files or data objects remain preserved [16][23].

2. Security framework for digital objects in the proposed environment

In this section we propose our security approach. In section 2.1 we present and discuss two known security models for enforcing integrity – the Biba model [1] and the Clark-Wilson model [4] – and in section 2.2 our Syntactic and Semantic model for verifying integrity introduced in [14] applicable for our requirements in the iRODS environment. At next we describe our novel approach of combining the chosen integrity model with our Syntactic and Semantic verification model and how this can be actually used for objects in section 2.3 and for collections or the archive as whole in section 2.4. Section 2.5 treats security for rules and micro-services and its limits for both.

This section mostly addresses integrity, as authenticity can be ensured by using mechanisms that satisfy the requirements explained in section 1.3, e.g. by using audit trails or version management.

2.1 Security models

Initially a security model which can realize the chosen policy of integrity must be selected. Two common integrity models are the Biba model [1] and the Clark-Wilson model [4]. Both are mandatory access controls, which means the

right of a subject to perform an action (open, read, write, …) upon an object is granted or denied by the system according to certain rules, in contrast to a discretionary access control where the rights depend only on the subject or the group it belongs to.

2.1.1 Summary of the Biba model

In the Biba model [1] a certain integrity level is assigned to the objects to be secured and the subjects who access the objects. A subject must not write to objects with a higher integrity level to avoid downgrading their integrity and must not read objects with a lower integrity level to not downgrade its own integrity level. Considering our environment and its purpose this integrity model has several drawbacks. In order to employ the Biba model each subject and object must be given an integrity level, which proves to be very difficult in practice. A subject is every user but also every process and therefore in our case every micro-service, too. An object is, of course, every archival object, including all metadata and derived informations, but also all users, rules and micro-services. Users are also objects because we may need to know the user credentials and thus read them or we do need to change his membership of a user group. For invoking rules we need to know their invocation conditions, the rules and micro-services of the work-flow chain as well as the rules and micro-services of the recovery chain. Each of these must be given an integrity level w.r.t. to the integrity levels of other subjects and objects. These integrity levels must be chosen such that they are not too low to deny write access to required objects, but not too high to deny read access to needed objects. Thus the Biba model can be very complex to implement with all specifications satisfied.

2.1.2 Summary of the Clark-Wilson model

In the Clark-Wilson model [4] objects are divided into objects that need to be in a valid state (i.e. integer), so called Constrained Data Items (CDI), and objects that need not, so called Unconstrained Data Items (UDI). The integrity of a CDI is verified by an Integrity Verification Procedure (IVP). UDIs or CDIs are transformed into a valid state as new CDI by a Transformation Procedure (TP). A TP must only be executed on a certain set of CDIs. If a TP accepts UDIs, it must be able to transform all possible values of an UDI or do no transformation at all. As seen in Table 1, the Clark-Wilson model fits very well to the iRODS environment.

Table 1. Mapping between Clark-Wilson entities and iRODS/SHAMAN entities

Clark-Wilson entity	iRODS/SHAMAN entity
UDI	Submission Information Package
CDI	Archival object/Archival Information Package, Dissemination Information Package, metadata, audit trails, users, (rules, micro-services)
IVP	Micro-services
TP	Rules, micro-services

The TP restriction for its executions can be modeled with the rule conditions chosen as such that they are only invoked for certain objects.

Our idea is to use IVPs for the validation of certain CDIs based on the Syntactic and Semantic model introduced in [14], summarized in the next section.

As this model is independent of the Clark-Wilson model and not all types of CDIs may be checked for integrity with this model, we simply use the term object.

2.2 Syntactic and Semantic model

In the preservation community there exists the common view that not just the objects themselves should be preserved but also their structural and semantic informations [23]. Objects are thus expanded into a concept of a digital entity which according to [10] consists of a) data, the bits saved to the media, b) Information, semantics tags that are appended to the bits, and c) knowledge, any relationship defined between the semantic tags.

For security, and in our case integrity and authenticity, this means that both must also include the structure and semantic of the objects or, in other words, that the described digital entity must be integer and authentic.

Thibodeau proposes in [19] a model that classifies objects in 3 levels: a physical level for the data medium, a logical level and the conceptual level. All 3 levels interact with each other as the bitstream represents a text document, image, audio or video file and is read from and written to the media. The advantage is that a change on one level does not necessary causes a change on another level, e.g. if an image is converted from JPEG to TIFF both differ at the physical and logical levels, but not on the conceptual level, because by an human they are perceived as the same. Thus one can say they are integer on the conceptual level, although not on the physical and logical levels. The other way is also possible: an alteration at the conceptual level does not necessary cause a

change on the other two levels, e.g. certain objects may be textually annotated, but a few decades later the used words have another meaning, which is a change on the conceptual level only. Such changes should also be measurable, e.g. in this case by using an ontology.

Our model proposed in [14] is based on Thibodeau's model, but extends his model by adding a semantic domain which is also structured into three levels. An object therefore consists of a syntactic and semantic domain, each having 3 levels.

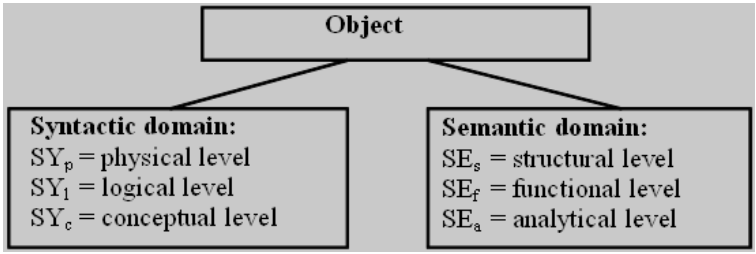

Figure 1. Levels of the Syntactic and Semantic model (modified from [14])

The physical level and logical levels are the same as in Thibodeau's model, the conceptual level is narrowed to the signal, the structural level is related to the presentation, the functional to the content and the analytical to the perception. The Syntactic and Semantic model of a object *O* consisting of all 6 items is described in the following so called "verifier tuple" [15]:

$O = \{SY_p, SY_l, SY_c, SE_s, SE_f, SE_a\}$

Each tuple item is a set of features of the respective level. To check the integrity of a certain level, all features of the feature set of the tuple item of the level must be checked. Sample features are listed in section 3.

As already stated, a loss of integrity at one level does not necessarily affect other levels. Therefore the verifier tuple can be expressed as seen in Figure 2.

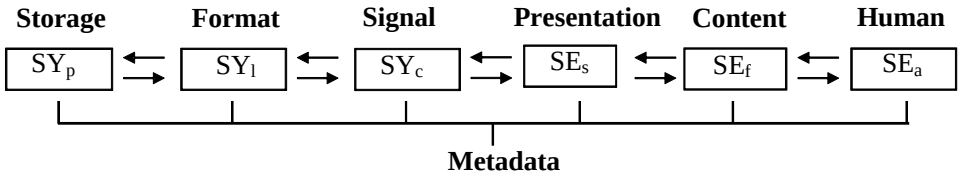

Figure 2. Syntactic and Semantic model (modified from [14])

The advantage of the verifier tuple is that a loss of integrity can be directly traced to the cause, with some restrictions:

As the analytical level is high-level semantics and directly related to the human perception it is very difficult to automatically extract the required

features. Therefore this level cannot completely proven to be integer. But as also all lower levels are affected, the integrity loss itself is detected, although it may not be always possible to discriminate integrity corruptions between functional and analytical level.

Additionally, inside a preservation system it may be difficult to directly extract the features on the physical level, e.g. the magnetization on hard disks, the lands/pits on CDs, due to limited access. But nevertheless it is sufficient to say an object is integer as long as the bits on the logical level were not altered.

Our model detects the location of losses of integrity and where they occurred. To ensure the integrity, losses must not only be detected but also repaired. An important part of this recovery are invertible TPs. iRODS was already prepared for this, as for every rule (TP), a recovery chain must be defined, which is executed in case of an error. Even if micro-services and rules are used in the workflow chain, which are not directly invertible, the state of the invalid objects can nevertheless be transitioned back to the former valid state using audit trails (a detailed log of changes) or at least using version management.

The next sections explain our idea for combining the Syntactic and Semantic model with the Clark-Wilson integrity model for objects (section 2.3), collections of these (section 2.4) and rules and micro-services (section 2.5) in the iRODS environment.

2.3 Verification of objects in iRODS environment

To verify an object the values for all syntactic and semantic features must be retrieved, where applicable, i.e. some or all values for the features of the physical and analytical levels are empty. Each object has metadata that are themselves objects (as listed in Table 1) and therefore must be verified, too.

A trivial approach to detect a loss of integrity is as follows: When an object is created in the system and every time it is changed its syntactic and semantic features must be calculated or updated and stored. An IVP would load this features and compare it the stored ones. An object is integer if both feature sets are equal and if its metadata (and the metadata of the metadata) is also integer. This trivial approach is very resource-consuming, as the feature sets may have dozens or even hundreds of features of non-atomic values. These must be stored in the archive and each item must be compared to its respective item and an IVP must ensure that every CDI is valid/integer.

For better resource usage, our suggestion is to hash the features. Therefore the hash of the reference features is stored in the archive and IVPs compare it against the calculated actual hash. Hash functions are divided into cryptographic hash functions and perceptual hash functions [20]. Cryptographic hash functions yield a different result if the message was changed by just one bit,

whereas a perceptual hash function is more fault-tolerant. Which one of the both should be chosen, depends on the feature as well of the level of integrity defined in the security policy. Features of the syntactic domain mostly use cryptographic hash functions, whereas features of the semantic domain rather employ a perceptual hash function. Furthermore, as the objects are in a hierarchy, we propose the usage of hierarchical hash functions.

In digital preservation, often there exist multiple replicas for digital objects, e.g. as backup, for comparison to ensure integrity, to ensure availability or due to distribution across the grid for faster access. For the sake of authenticity these replicas must be recognizable as such even if they are bit-identical.

2.4 Verification of collections in iRODS environment

In section 2.3 an novel approach to verify objects was presented by comparing its stored hash to its actual hash. This fails if the actual object is missing. The stored hash may either be existent or missing, too. In the first case the archive may deduce it from the global archive audit trail if a) the corresponding object was either removed, but left behind its hash, or b) should be somewhere, i.e. is really missing. If the stored hash is also missing, the system may not even know something is missing if it does not control the global audit trail. These cases should not occur as the IVPs – implementations of our proposed Syntactic and Semantic model were applicable – prevent TPs from producing this invalid state. However, it may happen due to influences uncontrollable by the archive, such as a hard drive crashes, power outage, etc.

The verification of integrity of the whole archive is therefore not just verification of the integrity of all objects, but should also treat the whole archive like every object and thus our idea is to calculate a (hierarchical) hash for it as a whole. If a hierarchical hash is used and some object on any level becomes corrupted, the whole archive loses its integrity. The advantage is that without explicitly scanning every object for integrity, which may take a long time for maybe millions of objects in a preservation archive, only the archive as a whole must be considered. A drawback is that in case a loss of integrity occurred, this scan must be performed nonetheless.

To avoid checking each object we propose the insertion of one or more hierarchy levels between the archive as a whole as the root of the hierarchy and each object from so far as a leaf. Objects are most likely grouped anyway by one or more defined properties, e.g. all images depicting a drawing from a certain painter. Some of these collections can be treated as objects and thus form the additional hierarchy levels. On detection of a loss of integrity of the whole archive, the search space is radically reduced.

2.5 Verification of rules and micro-services in iRODS environment

Rules and micro-services in iRODS are a special case in the verification problem as they are executable and therefore actively change the archive. In our approach TPs and IVPs are to be implemented as micro-services and rules and both are key items in the Clark-Wilson model. In iRODS the micro-services exist as source code on the machine and are statically linked and compiled into the iRODS client. The rules are listed in a special file. Our idea thus is, if the client is not running their integrity can be proven by checking these files, e.g. after a migration of the system. If the client is running there is no way to prove that a micro-service is actually integer, as the IVP for checking micro-services (and maybe other objects) is a micro-service itself and thus may be corrupt, too. Therefore as the system cannot guarantee that TPs perform well-formed operations, the Clark-Wilson model requires that TPs and IVPs should be certified to be valid by a security officer. To prevent fraudulent use, this security officer must not have the right to execute them [2]. The certification should include an inspection of the source code and a test run on a non-productive machine to evaluate its behavior.

3. Exemplary feature catalogue

In this section we give an example for the verification of an image CDI object, based on the features listed in Table 2. In this list, modified taken from [14], the features are sorted by their syntactic and semantic domain and are by no means complete. In [14] also feature lists for audio and handwriting are given, but our approach is not limited to those media types.

Table 2. Classification of image features

Syntactic domain	
Physical level SY_p	bp_1=storage characteristics
	bp_2=hard drive sector
	bp_3=memory segment
	bp_4=Lands/pits
	bp_i=…

Semantic domain	
Structural level SE_s	bs_1=input devices (sensor, camera, …)
	bs_2=A/D converter
	bs_3=program for output/application/interpretation (viewer)
	bs_4=output medium
	bs_5=operations and processes for archiving
	bs_i=…

Logical level SY_l	bl_1=data format (BMP, JPEG, TIFF, …)
	bl_2=bits per sample
	bl_3=channels
	bl_4=data stream
	bl_5=dimensions
	bl_i=…

Functional level SE_f	bf_1=histogram
	bf_2=object shapes
	bf_3=object positions
	bf_4=point of view
	bf_5=lumination
	bf_i=…

Conceptual signal level SY_c	bk_1=optical signal (wave lengths, …)
	bk_2=electrical signal (frequencies, magnitudes phases)
	bk_i=…

Analytical level SE_a	ba_1=object type, semantic
	ba_2=author, creator
	ba_3=origin
	ba_i=…

An IVP may be implemented as being responsible for all features or specialized for each domain or level. As described in section 2.3, our idea is after a CDI was transformed with a TP, e.g. an image processing operation the IVP responsible for this CDI is called. This IVP extracts the features compares the calculated hashes of these features with the stored hashes. If the IVP detects a change not tolerable by the policy, it reverts the objects into a valid CDI. Additionally, this occurrence is logged along with the information on which level or even which feature failed and should also trigger IVPs responsible for the integrity of these collections to which the object belongs.

4. Summary and Conclusion

We have shown how the integrity and authenticity of objects (archival objects, collections, archive as a whole) in a digital long-term preservation environment can verified and preserved. After introducing digital preservation, integrity, authenticity and the preservation system, we proposed a novel combination of the Clark-Wilson model as a security model for enforcing integrity as it fits best for our environment with the Syntactic and Semantic model for verification, whereby it not only allows the system to detect a loss of integrity later on, but also where it occurred. For the actual verification we proposed a hierarchical top-down approach, where a loss of information can be detected on the level of the archive as whole. The invalid features of an object can than be found with a decreased complexity by descending first into the invalid collections, sub-collections, the object, sub-objects (e.g. metadata) and the features themselves.

This contribution only presents a first study as the environment is currently developed. Our current work is the extension of the feature list, an examination how exactly the different features should be verified, i.e. what features can be combined into a common hash value, what hash methods should be used, and the implementation of these as IVPs.

Acknowledgements

The work in this paper has been supported in part by the European Commission through the FP7 ICT Programme under Contract FP7-ICT-216736 SHAMAN. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

[1] BIBA, K. J., *Integrity Considerations for Secure Computer Systems*, MTR-3153, The Mitre Corporation, April 1977.

[2] BISHOP, M., *Introduction to Computer Security*, Addison Wesley, 2004.

[3] CHOKHANI, S., FORD, W., SABETT, R., MERRILL, C., WU, S., *Request for Comments (RFC) 3647: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, NOV. 2003.

[4] CLARK, D. D., WILSON, D. R., *A Comparison of Commercial and Military Computer Security Policies*, 1987 IEEE Symposium on Security and Privacy, 1987.

[5] CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS (CCSDS), *Reference Model for an Open Archival Information System (OAIS). Recommendation for Space Data System Standards, CCSDS 650.0-B-1, Blue Book*, January 2002 (*http://public.ccsds.org/publications/archive/650x0b1.pdf*).

[6] DITTMANN, J., WOHLMACHER, P., NAHRSTEDT, K., *Multimedia and Security – Using Cryptographic and Watermarking Algorithms*, IEEE MultiMedia, October-December 2001, Vol. 8, No. 4, pp. 54-65, ISSN 1070-986X, 2001

[7] IRODS WEBSITE, *http://www.irods.org*

[8] KILTZ, S., LANG, A., DITTMAN, J., *Taxonomy for Computer Security Incidents*, In: Cyber Warfare and Cyber Terrorism; Publisher: Information Science Reference (IGI Global); L.J. Janczewski, A.M. Colarik (eds.); ISBN 978-1-59140-991-5; 2007

[9] KUNY, T., *A Digital Dark Ages? Challenges in the Preservation of Electronic Information*, International Preservation News 17, pp. 8-13, 1998 (*http://www.ifla.org/IV/ifla63/63kuny1.pdf*).

[10] MOORE, R., *The San Diego Project: Persistent Objects. Proceedings of the Workshop on XML as a Preservation Language*, Urbino, Italy, October 2002.

[11] NATIONAL DIGITAL INFORMATION INFRASTRUCTURE AND PRESERVATION PROGRAM WEBSITE, *http://digitalpreservation.gov*

[12] NESTOR, *Catalogue of Criteria for Trusted Digital Repositories – Version 1*, *http://edoc.hu-berlin.de/series/nestor-materialien/8/PDF/8.pdf*

[13] NESTOR WEBSITE, *http://www.langzeitarchivierung.de*

[14] OERMANN, A., DITTMANN, J., DOBRATZ, S., *Sicherung der Integrität und Authentizität in Digitalen Langzeitarchiven*, Proceedings of D-A-CH Security 2008, Berlin, Germany, June 24-25, 2008.

[15] OERMANN, A., LANG, A., DITTMANN, J., *Verifier-Tuple for Audio-Forensic to Determine Speaker Environment*, In: City University of New York: Multimedia and Security, MM&Sec'05, Proceedings, New York, NY, ACM, pp. 57-62, 2005.

[16] PHELPS, T. A., WATRY, P. B., *A No-Compromises Architecture for Digital Document Preservation*, Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005), September 18-23, 2005 Vienna, Austria.

[17] ROßNAGEL, A., SCHMÜCKER, P. (EDS.), *Beweiskräftige elektronische Archivierung – Bieten elektronische Signaturen Rechtssicherheit?*, Economica MedizinRecht.de Verlag, 2006.

[18] SHAMAN WEBSITE, *http://www.shaman-ip.eu*

[19] THIBODEAU, K., *Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years*, Council on Library and Information Resources: The State of Digital Preservation: An International Perspective, 2002.

[20] VOLOSHYNOVSKIY, S., KOVAL, O., BEEKHOF, F., PUN, T., *Robust perceptual hashing as classification problem: decision-theoretic and practical considerations*, Paper presented at the Proceedings of the IEEE 2007 International Workshop on Multimedia Signal Processing, Chania, Crete, Greece.

[21] WALLACE, C., PORDESCH, U., BRANDNER, R. *Long-term Archive and Notary Services (LTANS) - Long-Term Archive Service Requirements*, *http://ltans.edelweb.fr/draft-ietf-ltans-reqs-03.html*, 2004.

[22] WALLACE, C., PORDESCH, U., BRANDNER, R. *Request for Comments (RFC) 4810: Long-Term Archive Service Requirements*, March 2007.

[23] WATRY, P., *Digital Preservation Theory and Application: Transcontinental Persistent Archives Testbed Activity*, The International Journal of Digital Curation 2, 2007.