USING THE GLITE MIDDLEWARE TO IMPLEMENT A SECURE INTENSIVE CARE GRID SYSTEM.

Jesus Luna* , Marios D. Dikaiakos and Harald Gjermundrod
Department of Computer Science. University of Cyprus. PO Box 1678. Nicosia, Cyprus
jluna@cs.ucy.ac.cy
mdd@cs.ucy.ac.cy
harald@cs.ucy.ac.cy

Michail Flouris† , Manolis Marazakis and Angelos Bilas‡
Institute of Computer Science (ICS). Foundation for Research and Technology - Hellas (FORTH) PO Box 1385. GR-71110. Heraklion, Greece.
flouris@ics.forth.gr
maraz@ics.forth.gr
bilas@ics.forth.gr

Abstract Storage capabilities in novel "Health Grids" are quite suitable for the requirements of systems like ICGrid, which captures, stores and manages data and metadata from Intensive Care Units. However, this paradigm depends on widely distributed storage sites, therefore requiring new security mechanisms, able to avoid potential leaks to cope with modification and destruction of stored data under the presence of external or internal attacks. Particular emphasis must be put on the patient's personal data, the protection of which is required by legislations in many countries of the European Union and the world in general.

In a previous paper we performed a security analysis of ICGrid, from the point of view of metadata and data, where we found the need to protect the data-at-rest from untrusted Storage Elements (SE). That research also proposed a privacy protocol to protect a patients' private metadata and data.

This paper is the follow-up of our previous research, proposing an architecture based on gLite middleware's components, to deploy the contributed privacy protocol. As a proof of concept we show how to implement a Mandatory Access Control model for the metadata stored into the AMGA service. To protect the data itself, this paper presents our first experimental results on the performance

*This work was carried out for the CoreGRID IST project n°004265, funded by the European Commission.
†Also with the Dept. of Computer Science, University of Toronto, Toronto, Ontario M5S 3G4, Canada.
‡Also with the Dept. of Computer Science, University of Crete, P.O. Box 2208, Heraklion, GR 71409, Greece.

that can be achieved with a prototyped "cryptographic" Storage Resource Manager -CryptoSRM- service. Obtained results show that encrypting and decrypting at the CryptoSRM, instead of doing these at the SE or even at the Grid client, not only improve overall security, but also exhibit a higher performance that can be further improved with the aid of specialized hardware accelerators.

Keywords: Cryptography, gLite, Intensive Care Grid, privacy, security.

1. Introduction

Modern eHealth systems require advanced computing and storage capabilities, leading to the adoption of technologies like the Grid and giving birth to novel *Health Grid* systems. In particular, Intensive Care Medicine uses this paradigm when facing a high flow of data coming from Intensive Care Unit's (ICU) inpatients. These data needs to be stored, so for example data-mining techniques could be used afterwards to find helpful correlations for the practitioners facing similar problems. Unfortunately, moving an ICU patient's data from the *traditionally isolated* hospital's computing facilities to Data Grids via public networks (i.e. the Internet) makes it imperative to establish an integral and standardized security solution, harmonized with current eHealth Legislations, and able to avoid common attacks on the data and metadata being managed.

In our previous research related with the security analysis of Grid Storage Systems [19] we concluded that current technological mechanisms were not providing comprehensive privacy solutions and worst of all, several security gaps at the Storage Elements were found to be open. In an effort to cover these security gaps, the second part of our research [20] contributed with a *low-level* protocol for providing privacy to current Intensive Care Grid systems from a data-centric point of view, but taking into account the legal framework and keeping compliance with *high-level* mechanisms (i.e. the Electronic Health Card [22]). The contributed protocol proposed the use of a cryptographic mechanism, co-located with the the Storage Resource Manager (SRM [24]), to enhance a patient's data confidentiality. A second mechanism based on data-fragmentation was also proposed by our research to benefit data's assurance and overall performance. The latter mechanism has been investigated in [18].

Due to performance concerns, this paper presents an architecture for implementing the cryptographic mechanisms of the proposed privacy protocol, using components from the gLite middleware [7], and applying it to the ICGrid [15] system's data. As a proof of concept we present our first results on "the cost of security", that is, a performance comparison among a commonly used security approach (data encryption and decryption at the Grid client) and the proposed privacy protocol (data encryption and decryption at a central *cryptoSRM*). Also this paper contributes with a proposal for protecting ICGrid's metadata via a

Mandatory Access Control model in AMGA [26], to enforce different levels of authorization to the patient's personal information, thus fulfilling current eHealth Legislations.

The rest of this paper is organized as follows: Section 2 reviews the basic terminology related with the ICGrid system, along with the privacy issues that appear in the eHealth context. Section 3 describes a gLite-based middleware architecture required to implement the proposed privacy protocol for ICGrid. Our first experimental results on the cryptographic performance achieved by our proposal are shown in Section 4. Section 5 briefly presents the State of the Art related with our research. Finally, Section 6 presents our conclusions and future work.

2. The ICGrid system

In this Section we present the required data and security background of the ICGrid system studied in this paper.

2.1 Data and metadata architecture

Although a number of dedicated and commercially available information systems have been proposed for use in Intensive Care Units (ICUs) [13], which support real-time data acquisition, data validation and storage, analysis of data, reporting and charting of the findings, none of these systems was appropriate in our application context. Another important issue with ICU is the need for data storage: an estimate of the amount of data that would be generated daily is given in the following scenario. Suppose that each sensor is acquiring data for storage and processing at a rate of 50 bytes per second (it is stored as text) and that there are 100 hospitals with 10 beds each, where each bed has 100 sensors. Assuming that each bed is used for 2 hours per day, the data collected amounts to 33.5275 GB per day. But this number only represents the data from the sensors. Additional information includes metadata, images, etc. Because Grids represented a promising venue for addressing the challenges described above, the Intensive Care Grid (ICGrid) system [15] has been prototyped over the EGEE infrastructure (Enabling Grids for E-sciencE [1]). ICGrid is based on a hybrid architecture that combines a heterogeneous set of monitors that sense the inpatients and three Grid-enabled software tools that support the storage, processing and information sharing tasks.

The diagram of Figure 1 represents a Virtual Organization of the ICGrid system, which depicts the acquisition and annotation of parameters of an inpatient at an ICU Site (bottom left) and the transfer of data replicas to two *Storage Elements (SEs)*. The transfer comprises the actual sensor data, denoted as *Data*, and the information which is provided by physicians during the annotation phase, denoted as *Metadata*. We utilize the notion of a *Clinically*

Interesting Episode (CIE) to refer to the captured sensor data along with the metadata that is added by the physician to annotate all the events of interest.

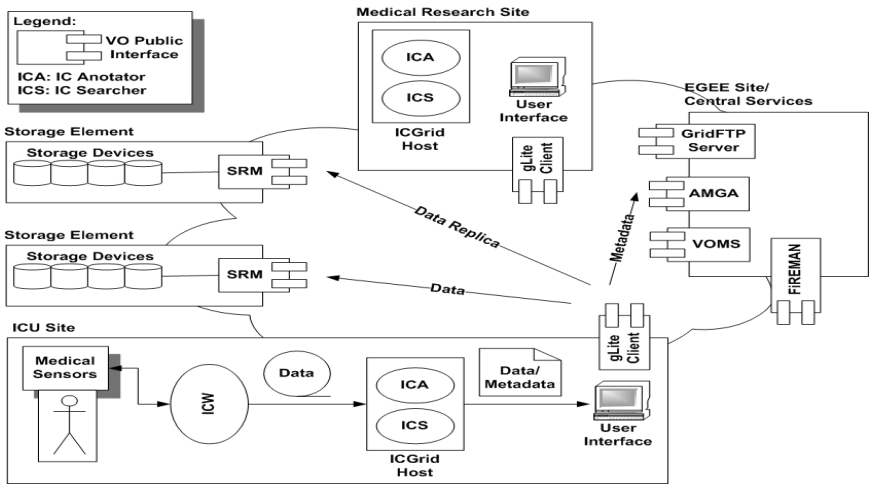

Figure 1. Architecture of an ICGrid's Virtual Organization.

When ICGrid's Data and Metadata are transferred to Storage Elements and Metadata servers (currently a gLite Metadata Catalogue -AMGA- service [26]) respectively, a set of messages are exchanged among the different entities. In particular we should highlight that file catalog services are being provided by FiReMAN (File Replication MAnager [12]) and, authorization mechanisms rely on the X.509 credentials issued by the Virtual Organization Membership Service (VOMS [9]).

2.2 Security and privacy issues

The deployment of production-level Health Grids, such as the ICGrid, should provide assurances of the patient's data, in particular when referring to personal information, which is currently the subject of increasing concerns in most legislations in the European Union [23]. Unfortunately, when personal data is being transferred from the Hospital to the Grid new vulnerabilities may appear: on the wire, at-rest, within the metadata servers, etc. A major concern in Health Grids is the adequate confidentiality of the individual records being managed electronically, which are usually stored as metadata. In the European Union, the patient's personal data is protected through the concept of *consent*, which can be interpreted as the freely given decision of the patient -or authorized party- to proceed with the processing of his personal data. Taking into consideration the legal framework and as a first step in proposing a pri-

vacy mechanism for the ICGrid, a previous paper [20] performed a security analysis of ICGrid's data and metadata by applying a framework previously extended and used in Grid storage services [19]. The results of the analysis have shown the need to protect the system from *untrusted Storage Elements*, which have full control over the stored information, thus allowing its leak, destruction of change due to successful external or even internal attacks. It is also worth highlighting that the mentioned analysis took into consideration the use of commonly deployed security mechanisms, namely the Grid Security Infrastructure [29] and the novel Electronic Health Card [22].

Based on the ICGrid's security analysis, the research presented in [20] also introduced a privacy protocol able to provide a well differentiated protection to the patient's data and metadata. The contributed protocol proposed the use of the gLite middleware [7] not only to provide data confidentiality, but also integrity, high availability and a privacy mechanism for the metadata, keeping compliance with the legal and technological aspects widely discussed in [10].

3. Secure ICGrid: protecting Metadata and Data

In this section we will present the main components of an architecture proposed to provide security to the ICGrid system introduced in Section 2. The specific goal of our proposal is to avoid data and metadata attacks (leakage, change or destruction) while at-rest into the untrusted Storage Elements. It is worth noticing that performance issues related with the cryptographic mechanism have been carefully considered in our design (more about this in Section 4). Because our previous security analysis [20] found that ICGrid's metadata and data require different security policies, the enforcement mechanisms presented in this section implement a differentiated approach for metadata (Section 3.2) and data (Section 3.3).

3.1 Architecture

Based on ICGrid's current architecture (figure 1), our proposal contributes with the following *Privacy Services*, co-located with the Central Services (scoped at the Virtual Organization level) and interacting directly with the GridFTP Server [14] and AMGA:

- CryptoSRM: This component is a modified Storage Resource Manager that apart from implementing the interface defined in [24], uses a cryptographic engine for encrypting and decrypting staged data stored in its local cache.

- Hydra Key Store: Implements a secure repository for the encryption keys [3]. The repository itself uses a fragmentation algorithm [25] for

providing confidentiality and high-availability to the cryptographic material.

- Secure Log: A secure logging service may help to back-trace potential abuses (even those performed by Grid administrators colluded with attackers).

3.2 Metadata Security

AMGA stores metadata in a hierarchical structure that resembles a Unix File System, and also its native authorization model is based on Access Control Lists [5] with POSIX-like permissions per-entry and directory (*r*=read, *w*=write and *x*=change into directory) and, an additional "admin flag" allowing users in a group to administer the ACLs of an entry. Using the latter mechanism, we have defined an authorization model for ICGrid's metadata based on the Bell-LaPadula Model's Mandatory Access Control (MAC) rules [11]:

1 The *Simple Security Property* states that a subject at a given security level may not read an object at a higher security level (no read-up).

2 The **-Property* (read star-property) states that a subject at a given security level must not write to any object at a lower security level (no write-down) and, may only append new data to any object at a higher security level.

Bell-Lapadula's Model applied to ICGrid's metadata (implemented over AMGA) can be seen in figure 2. The proposed MAC model is able to provide a basic level of confidentiality to the patient's private metadata, while at the same time "protecting" him from accidentally disclosing this information to the lower-security levels. In this example we have defined three different players (Patient -owner-, Paramedics -group- and the Intentive Care Unit Receptionist -others-) and also, three levels of authorization (Public, Semi-Private and Private). With the proposed AMGA's permissions on directories and entries it is possible to achieve the following Mandatory Access Control:

- Public Metadata: both Patient and Paramedics can read the entries, but only the ICU Receptionist can read and write them (i.e. schedule a new appointment with the physician).

- Semi-Private Metadata: the Paramedics can read and write entries (i.e. emergency information), the ICU Receptionist can only append new ones (the Paramedics group requires the admin flag to set read-only permissions to these newly created entries) and, the Patient is only able to read this metadata.

- Private Metadata: This is the most confidential level of the metadata, therefore only the Patient has full control over it (administrative permissions are implicit since he is the owner of his directories), while Paramedics and ICU Receptionists only can append new entries (the Patient must manage permissions of these newly created entries).

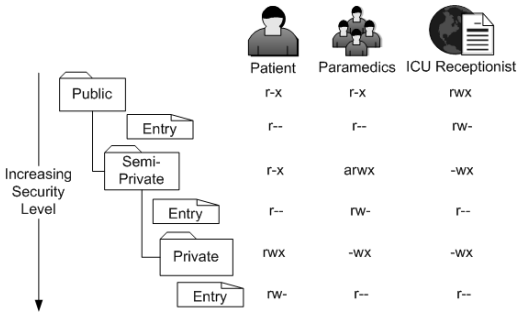

Figure 2. Mandatory Access Control model for ICGrid's Metadata.

Enforcing the *-Property's append-only mode conveys an administrative overhead for both, Patients and Paramedics, which must manage permissions for entries being created by lower-security subjects. Also it is worth to notice that native AMGA's authorization mechanism can not prevent a malicious System Administrator from accessing the metadata of all the stored patients. To cope with these issues, our future work considers the use of cryptographic techniques to provide greater confidentiality and even a consent-like mechanism (based on electronic signatures) to AMGA's metadata. This research will be briefly introduced in Section 6.

3.3 Data Security

Using the Privacy Services discussed in Section 3.1 it is possible to improve overall security and privacy using cryptography. Figure 3 shows how the different Privacy Services interact with the Central Services when an IC Annotator (ICA) stores data into the ICGrid system. In this figure we use the file naming notation from [16], when referring to the data being managed by the Grid: *(i)* Logical File Name -LFN- (a human readable identifier for a file), *(ii)* Global Unique Identifer -GUID- (a logical identifier which guarantees its uniqueness by construction) and, *(iii)* Site URL -SURL- (specifies a physical instance of a file replica, which is accepted by the Storage Element's SRM interface).

The core of our proposal is the CryptoSRM, which is responsible for symmetrically encrypting the staged data, previously transferred via a *secure channel* by the ICA's GridFTP client. Afterwards the encryption key is securely

stored in the Hydra service and the encrypted data moved to the untrusted Storage Element. It is obvious that attackers colluded with the latter will be unable to recover the original clear-text. A second scenario (Figure 4) considers an IC

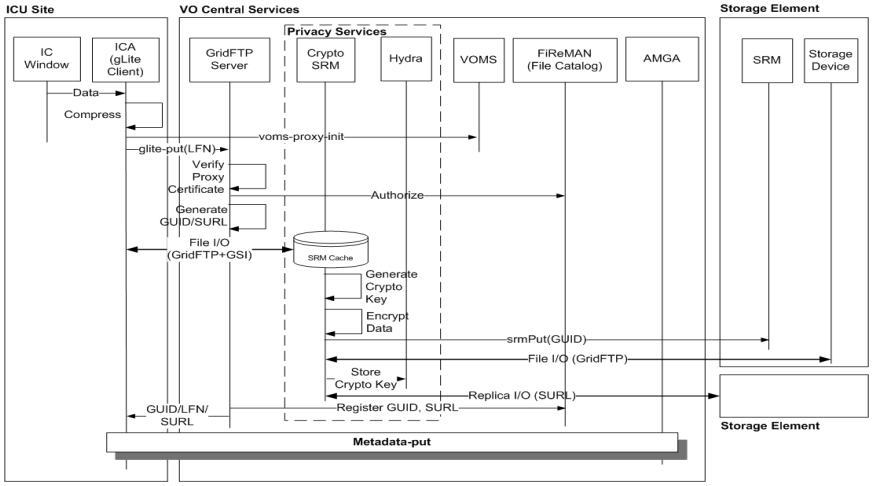

Figure 3. Secure ICGrid: transferring data.

Searcher (ICS) retrieving data from the ICGrid: in this case the encrypted data is transferred from the Storage Element, decrypted at the CryptoSRM (the appropriate key is obtained from Hydra) and conveyed through a secure channel to the ICS' GridFTP client. *Notice that the encryption key is never disclosed to the ICS, therefore avoiding its leak by potential attackers (i.e. reading the DRAM like in [6]).* A more comprehensive analysis of the performance issues related with our proposal is presented in the next section.

4. Experimental Results

We have setup the following testbed to measure the expected performance to be achieved with the protocol proposed in Section 3.3:

- Grid client (GC): this CentOS4-based node has been configured as a "gLite User Interface". It is an IBM xSeries 335, with two Intel Xeon HT processors @ 2.8GHz and 2GB of RAM.

- Storage Element (SE): To simulate the basic functionalities of the proposed CryptoSRM, we have used for the tests a "DPM_mysql Storage Element" running over Scientific Linux version 3.09. The SE uses a Dell PowerEdge1400, with two Intel Pentium III processors @ 800MHz and 784MB of RAM.

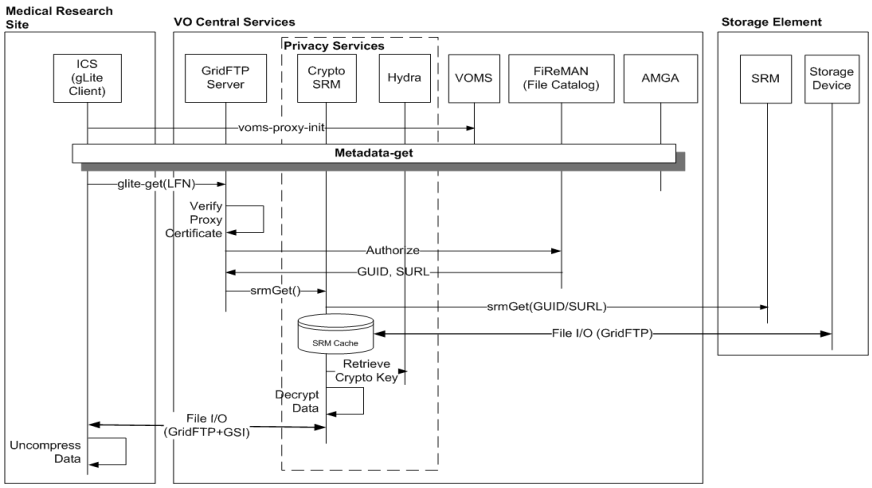

Figure 4. Secure ICGrid: retrieving data.

For the Data, random samples corresponding to one day of ICGrid's operation were generated for *(i)* a sensor (approx. 352 Kb), *(ii)* a bed (approx. 35157 Kb) and, *(iii)* a Hospital (approx. 351563 Kb). The *gzip* utility is used with its default parameters for compression, while for encryption the *aes-128-cbc* algorithm from the OpenSSL library (version 0.9.8g) was used. For comparison purposes we have measured the protocol's performance as the User's time (reported by the Unix *time* command) consumed by each phase of the following scenarios:

1 Grid client Encryption: This approach performs encryption/decryption at the Grid client and is commonly used by existing solutions (see Section 5). The steps taking place are: data compression, encryption and transfer to the SE via clear-text FTP. The inverse sequence is used to retrieve it from the SE.

2 CryptoSRM Encryption: This scenario simulates the basic steps proposed by our protocol: data compress, transfer via a GSIFTP encrypted channel to the CryptoSRM and finally, encryption at this entity. The inverse sequence of steps is used to retrieve stored data from the simulated CryptoSRM.

Each test was repeated 50 times to isolate potential overhead being caused by other processes concurrently running at the server. Table 1 shows how the size of the three data samples changed after the compression and encryption processes. It is worth to notice that the compressed data's size is about 60% of

the original one, however after encryption the size incremented approximately 35% for all the cases.

Table 1. Reported sizes (in KB) for the three ICGrid's Data Samples after compression and encryption

Data Sample	Original	Compressed	Encrypted
Sensor	352	213	288
Bed	35157	21213	28726
Hospital	351563	212125	287258

Figures 5, 6 and 7 show the performance results using ICGrid's data mentioned in Table 1. The three figures show a side-by-side comparison of the Grid client encryption (the Sensor, Bed and Hospital graphs), versus the CryptoSRM encryption (the Sensor-Sec, Bed-Sec and Hospital-Sec graphs). Aggregated values for the tested scenarios are given by the *TOTAL UP* and *TOTAL DOWN* bars. Figure 5 shows the only case in which uploading and downloading Data

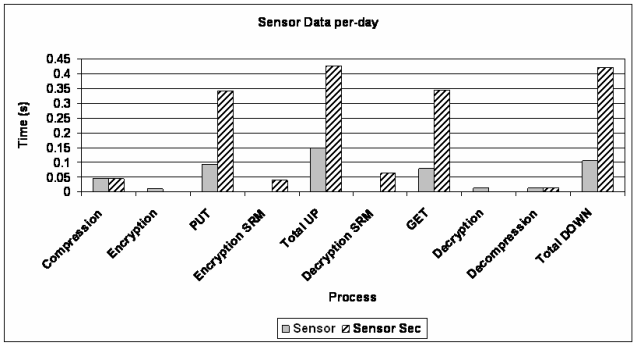

Figure 5. Processing a day of ICGrid's Sensor-data with the proposed privacy protocol.

through a secure GSI channel (the PUT and GET Sensor_Sec graphs), took more time than its equivalent via a clear-text FTP channel. This could be related to the GSI-transfer protocol itself, which for small data sizes requires more processing time (i.e. for encryption or padding). On the other hand for bigger data sizes, the performance achieved when uploading the Bed and Hospital Data (figures 6 and 7) is slightly less with the proposed privacy protocol (between 3%-4%) than with the Grid client encryption. This is because the data's size being uploaded to the SE is *smaller* in clear-text than when encrypted (around 30% according to Table 1), this latter fact helped to masquerade the overhead caused by the SE encryption mechanism (which provided

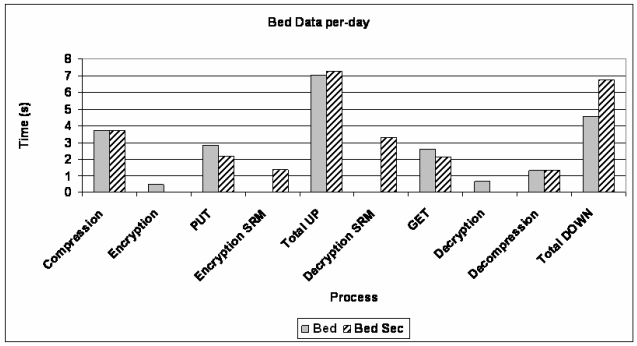

Figure 6. Processing a day of ICGrid's Bed-data with the proposed privacy protocol.

approx. 20% of the TOTAL UP time). When downloading Data the overall performance of the proposed protocol was about 39%-47% less than that of the Grid client encryption, however we have found that most of this overhead is due to the decryption operation taking place at the SE (which spent around 45% of the TOTAL DOWN time). This behavior was predicted, as the used SE is more biased towards storage than processing (this can be easily seen by comparing its hardware configuration with that of the Grid client). Despite this configuration, the experimental results have shown the viability of using the proposed CryptoSRM and it can be foreseen that if both, the SE and the Grid client, would have at-least the same hardware configuration, then for the Hospital's Data our proposal would improve with about 17% for the TOTAL UP time, and approximately with 11% for the TOTAL DOWN time of the Grid client-based encryption approach.

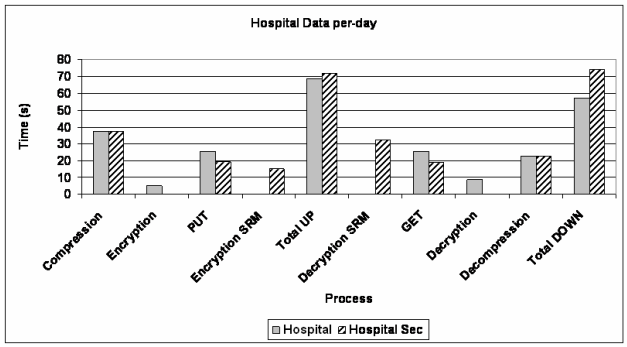

Figure 7. Processing a day of ICGrid's Hospital-data with the proposed privacy protocol.

5. Related Work

Nowadays most of the work related with Health Grids' security and privacy focuses on "high-level" authentication and authorization mechanisms that rely on Grid-IDs and VOMS-like infrastructures [9], therefore leaving data vulnerable in the untrusted Storage Elements. An example of these mechanisms can be seen in the BRIDGES [27] and SHARE [4] Health Grids.

The research that is closely related with the work presented in this paper has been presented in [21], where the authors also used the gLite middleware to protect medical images. Their system ensures medical data protection through data access control, anonymization and encryption. A fundamental difference with our approach is the use of encryption at the Grid client, which requires retrieving the encryption key from a Hydra Keystore for decrypting the image. With our research it has been shown that such approach does not only introduce uncertainties about the key's confidentiality (it may be compromised at the Grid client), but also has a performance lower than our "centralized" proposal (using the CryptoSRM).

There are other state of the art distributed storage systems that, even though they have not been specifically designed for the Health Grid, they have focused on low-level data protection by implementing encryption mechanisms at the "Grid's edges" (therefore disclosing the encryption key to the untrusted SEs and Grid Clients). For example in OceanStore [17], stored data are protected with redundancy and cryptographic mechanisms. An interesting feature in OceanStore is the ability to perform server-side operations directly on the encrypted data, this increases system's performance without sacrificing security. On the other hand it is worth to mention the Farsite system [8], which provides security and high availability by storing encrypted replicas of each file on multiple machines.

A second group of related systems do not rely on cryptography, but in a "data fragmentation" scheme for data protection. In the first place let us mention POTSHARDS [28], which implements an storage system for long-time archiving that does not use encryption, but a mechanism called "probably secure secret splitting" that fragments the file to store prior to distributing it across separately-managed archives. A similar approach is given by Cleversafe [2] via an Information Dispersal Algorithm (based on the Reed-Solomon algorithm) for its open-source *Dispersed Storage Project*. In general both, POTSHARDS and Cleversafe, are interesting solutions that solves the management problems posed by cryptosystems and long-living data, however the security achieved only by fragmenting the files could not be strong enough for some highly-sensitive environments.

6. Conclusions

In this paper we have presented a follow-up to our research on data-level security for Health Grids. After analyzing in a previous work the security requirements of the proposed scenario, we found the need to protect Metadata and Data from untrusted Storage Elements and Grid Clients that could compromise sensitive material (i.e. cryptographic keys). The second part of this research proposed a privacy protocol to protect the patient's personal information (metadata) along with his data, using two basic mechanisms: encryption and fragmentation. This paper has proposed building the cryptographic mechanism using components from the gLite middleware, in particular the Hydra Keystore a Storage Resource Manager with encryption facilities (the CryptoSRM).

About the Metadata, this paper proposed the implementation of an Mandatory Access Control model via AMGA's access control lists. This model was inspired in the Bell-Lapadula's model and the Electronic Health Card, currently being deployed in the European Union. Despite its simplicity, the proposed approach enforces different levels of authorization for a patient's personal data, in compliance with the eHealth Legislations studied in our previous work. However, we still have a lot of work to do in Metadata confidentiality, because currently AMGA is not able to offer protection from malicious administrators with direct access to its database.

Management of Health Grid's Data has taken a different approach in our proposal, so as a proof of concept to justify –from a performance point of view– the use of a "centralized" encryption mechanism (the CryptoSRM), in this paper we have simulated the former with a SE able to encrypt Data coming from an ICGrid client. Data's transfer operations (upload and download) resulted in most of the protocol's overhead, therefore suggesting us to keep transferred Data as small as possible. Taking into account that the encrypted Data is greater in size than its clear-text counterpart, we highly recommend not performing encryption at the "edges" of the Grid (i.e. Grid client, Storage Element). Notice that this argument is fully compatible with our previous security analysis, which established that Storage Elements are untrusted, thus encryption keys should not be delivered neither to them or even to the Grid Clients. Despite the hardware configuration being used to simulate the CryptoSRM in our experiments, it was possible to conclude its viability for the proposed privacy protocol. We can foresee that an important improvement in overall security and performance can be achieved, if the CryptoSRM uses a hardware-based cryptographic-accelerator, future work should prove this point.

Even though we have shown that for ICGrid using the proposed CryptoSRM is feasible, we believe that a more general solution (i.e. for demanding HEP applications) may be willing to "sacrifize" security by moving the cryptographic mechanism to the Storage Elements, thus benefiting performance and scalabil-

ity. To cope with untrusted Storage Elements under such assumption, the next part of our ongoing research will also focus on the fragmentation mechanism proposed in [20], which benefits Data's availability and bandwidth use. We are planning to build analytical models, as those used in [18], to show the relationship between Data's assurance, Data's fragments and incurred overhead. A prototype using Cleversafe's API (Section 5) will be also developed for our test. Also as Future Work we are planning to study, along with AMGA's creators, the repercussions of using encryption at different levels of the Metadata.

Acknowledgments

The authors would like to thank EGEE-II (contract number INFSO-RI-031688) and Asterios Katsifodimos (University of Cyprus) for his technological support to perform the tests presented in this paper.

References

[1] Enabling Grids for E-SciencE project. http://www.eu-egee.org/.

[2] Cleversafe. http://www.cleversafe.com, 2007.

[3] Encrypted Storage and Hydra. https://twiki.cern.ch/twiki/bin/view/EGEE/DMEDS, September 2007.

[4] SHARE: Technology and Security Roadmap. http://wiki.healthgrid.org/index.php/Share_Roadmap_I, February 2007.

[5] Amga: Users, groups and acls. http://project-arda-dev.web.cern.ch/project-arda-dev/metadata/groups_and_acls.html, 2008.

[6] Disk encryption easily cracked. http://www.networkworld.com/news/2008/022108-disk-encryption-cracked.html, 2008.

[7] glite: Lightweight middleware for grid computing. http://www.glite.org/, 2008.

[8] Atul Adya, William J. Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R. Douceur, Jon Howell, Jacob R. Lorch, Marvin Theimer, and Roger Wattenhofer. Farsite: Federated, available, and reliable storage for an incompletely trusted environment. In *OSDI*, 2002.

[9] R. Alfieri, R. Cecchini, V. Ciaschini, L. dellÕAgnello and?A. Frohner, A. Gianoli, K. Lorentey, and F. Spataro. VOMS, an Authorization System for Virtual Organizations. In *First European Across Grids Conference*, February 2003.

[10] European Health Management Association. Legally eHealth - Deliverable 2. http://www.ehma.org/_fileupload/Downloads/Legally_eHealth-Del_02-Data_Protection-v08(revised_after_submission).pdf, January 2006. Processing Medical data: data protection, confidentiallity and security.

[11] D. Elliot Bell and Leonard J. LaPadula. Secure computer systems: A mathematical model, volume ii. *Journal of Computer Security*, 4(2/3):229–263, 1996.

[12] JRA1 Data Management Cluster. EGEE: FiReMAN Catalog User Guide. https://edms.cern.ch/document/570780, 2005.

[13] B.M. Dawant et al. Knowledge-based systems for intelligent patient monitoring and management in critical care environments. In Joseph D. Bronzino, editor, *Biomedical Engineering Handbook*. CRC Press Ltd, 2000.

[14] Open Grid Forum. GridFTP: Protocol Extensions to FTP for the Grid. http://www.ggf.org/documents/GWD-R/GFD-R.020.pdf, April 2003.

[15] K. Gjermundrod, M. Dikaiakos, D. Zeinalipour-Yazti, G. Panayi, and Th. Kyprianou. Icgrid: Enabling intensive care medical research on the egee grid. In *From Genes to Personalized HealthCare: Grid Solutons for the Life Sciences. Proceedings of HealthGrid 2007*, pages 248–257. IOS Press, 2007.

[16] JRA1. EGEE gLite User's Guide. https://edms.cern.ch/document/570643/, March 2005.

[17] John Kubiatowicz, David Bindel, Yan Chen, Steven E. Czerwinski, Patrick R. Eaton, Dennis Geels, Ramakrishna Gummadi, Sean C. Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Y. Zhao. Oceanstore: An architecture for global-scale persistent storage. In *ASPLOS*, pages 190–201, 2000.

[18] J. Luna et al. Providing security to the desktop data grid. Submitted to the 2nd. Workshop on Desktop Grids and Volunteer Computing Systems (PCGrid 2008).

[19] J. Luna et al. An analysis of security services in grid storage systems. In *CoreGRID Workshop on Grid Middleware 2007*, June 2007.

[20] Jesus Luna, Michail Flouris, Manolis Marazakis, Angelos Bilas, Marios Dikaiakos, Harald Gjermundrod, and Theodoros Kyprianou. A data-centric security analysis of icgrid. In *Proceedings of the CoreGRID Integrated Research in Grid Computing*, pages 165–176, 2008.

[21] Johan Montagnat, Ákos Frohner, Daniel Jouvenot, Christophe Pera, Peter Kunszt, Birger Koblitz, Nuno Santos, Charles Loomis, Romain Texier, Diane Lingrand, Patrick Guio, Ricardo Brito Da Rocha, Antonio Sobreira de Almeida, and Zoltan Farkas. A secure grid medical data manager interfaced to the glite middleware. *J. Grid Comput.*, 6(1):45–59, 2008.

[22] Federal Ministry of Health. The Electronic Health Card. http://www.die-gesundheitskarte.de/download/dokumente/broschuere_elektronische_gesundheitskarte_engl.pdf, Octuber 2006. Public Relations Section. Berlin, Germany.

[23] European Parliament. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal of the European Communities of 23 November 1995 No L. 281 p. 31., Octuber 1995.

[24] T. Perelmutov et al. SRM Interface Specification v2.2. Technical Report, FNAL, USA, 2002.

[25] Michael O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. ACM*, 36(2):335–348, 1989.

[26] N. Santos and B. Koblitz. Distributed Metadata with the AMGA Metadata Catalog. In *Workshop on Next-Generation Distributed Data Management HPDC-15*, June 2006.

[27] Richard O. Sinnott, Micha Bayer, A. J. Stell, and Jos Koetsier. Grid infrastructures for secure access to and use of bioinformatics data: Experiences from the bridges project. In *ARES*, pages 950–957, 2006.

[28] Mark W. Storer, Kevin M. Greenan, Ethan L. Miller, and Kaladhar Voruganti. Secure, archival storage with potshards. In *FAST'07: Proceedings of the 5th conference on USENIX Conference on File and Storage Technologies*, pages 11–11, Berkeley, CA, USA, 2007. USENIX Association.

[29] Von Welch. Globus toolkit version 4 grid security infrastructure: A standards perspective. http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf, 2005. The Globus Security Team.