pHealth 2021
B. Blobel and M. Giacomini (Eds.)

doi:10.3233/SHTI210609

Implementation of Privacy and Security for a Genomic Information System

Jaime DELGADO[1] [a], Silvia LLORENTE [a] and Guillem REIG [b]
[a] *Information Modeling and Processing (IMP) group – DMAG, Computer Architecture Dept. (DAC),*
[b] *inLab, Facultat d'Informàtica de Barcelona (FIB), Universitat Politècnica de Catalunya (UPC - BarcelonaTECH)*

Abstract. Genomic information is key for the implementation of real personalized medicine. Nevertheless, access to this kind of information must be controlled because of its high privacy and security requirements. Several genomic information formats exist, although we have started from MPEG-G as it includes metadata and protection mechanisms since its inception and provides a hierarchical structure to organize the information contained. The proposed GIPAMS modular architecture provides a secure and controlled access to genomic information, which may help on improving personalized medicine as described in this paper.

Keywords. Genomics, privacy, modular architecture, GIPAMS

1. Introduction

Personalized medicine is one of multiple examples of use of genomic information, which is a currently relevant research topic. In this context, genomic information sequencing and processing is gaining momentum. Sequencing price has decreased in the last years and now it is possible to sequence complete human genome for a thousand euros [1]. This is leading to an increase of information, making difficult its storage and management by different organizations.

On the other hand, genomic information should be protected from unauthorized access due to its specific characteristics, as it includes sensible information not only from a person but also her relatives. This means that, once the information is leaked, it cannot be revoked, like a certificate, and it is public "forever".

Nevertheless, protection mechanisms currently exist, but they have different limitations, such as being restricted for closed environments or being regulated by Data Access Committees, implying in this case a possible long and tedious process that may slow down reaching research results.

In order to provide a possible solution to privacy protection of genomic information, we present in this paper our proposal of a secure modular system, called GIPAMS (Genomic Information Protection And Management System), which defines mechanisms for privacy, protection, storage, search and access to genomic information. A first description of GIPAMS is given in [2]. We have implemented a first version of the

[1] Corresponding Author: Jaime Delgado, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain; e-mail: jaime.delgado@upc.edu

system. As information representation, we have chosen the file structure provided by ISO/IEC 23092, Genomic Information Representation (MPEG-G) [3][4]. The next subsection briefly introduces this structure.

In the rest of the paper we describe the proposed GIPAMS architecture, some details of its implementation with MPEG-G and some discussions on how it may help in the implementation of personalized medicine.

1.1. MPEG-G file structure

MPEG-G [3], currently developed and maintained by ISO/IEC JTC1 SC29/WG8, is a standard devoted to the representation of genomic information in a compressed and secure way, including metadata and protection features into the same file structure.

It is structured in hierarchical boxes, as represented in Figure 1. The boxes may contain data, like the File header (flhd) or subboxes, like Dataset Group (dgcn), which in turn may contain information boxes or container boxes, until the last level, which may be organized in access units (aucn) or in descriptor streams (dscn), depending on how the genomic information has to be accessed. In the end, genomic information is stored in blocks, regardless using access units or descriptor streams.

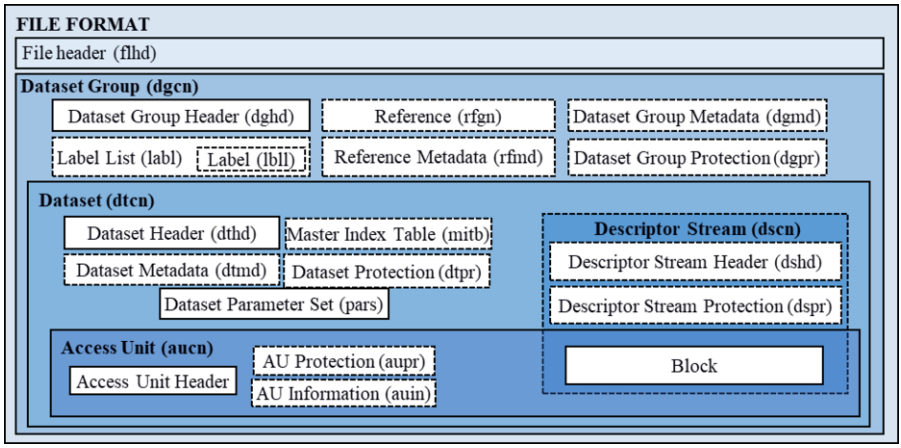

Figure 1. MPEG-G file format (adapted from MPEG-G standard).

2. Methods – GIPAMS Description

The architecture of the developed platform is an evolution of our original Multimedia Information Protection And Management System, MIPAMS [5]. As it now deals with genomic information, we have called it Genomic Information Protection And Management System, GIPAMS. Its structure is depicted in Figure 2. Not all the modules are currently implemented, but this is the expected final complete picture. The functionality of the different modules is as follows:

- User Application: Web application that sends requests to the Workflow Manager based on user actions. Communication between this application and the rest of the architecture is through a secure channel.

- Workflow Manager: Intermediate module that acts as a unique entry point to the system. It checks operation authorization before interacting with other modules.
- Authentication Service: User identification server, which uses OAuth 2.0 [6] and JSON Web Tokens [7]. Its implementation is currently based on Keycloak [8].
- Genomic Content Service: Module in charge of genomic archive management, both in reading and writing operations. It uses scripts as presented in 3.2.
- Authorization Service: Module for authorization rules validation.
- Search Service: Performs searches over genomic information. It provides extra filtering features with the use of a relational database.
- Policy Service: Module in charge of the creation of the authorization rules, which are organized into policies.
- Protection Service: Module which creates protection information metadata as well as applies the mechanisms defined (i.e. encryption, signature, etc.).
- Report / Track Service: Module in charge of reporting the operations done in the system, especially those not authorized.
- Certification Authority: Provides digital certificates to secure communications.

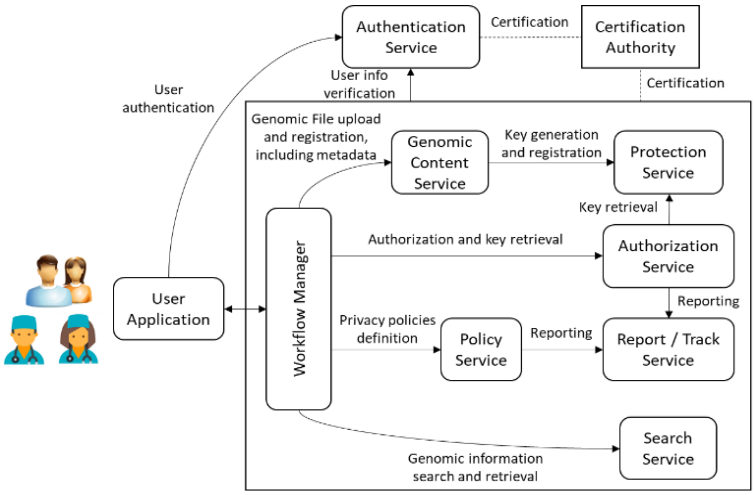

Figure 2. GIPAMS Architecture.

3. Results

When implementing GIPAMS with MPEG-G information, we need to deal with two relevant problems, first, integrating real metadata and, second, simplifying the file structure to facilitate development. We briefly describe how we have solved these problems in the next subsections.

3.1. Metadata Mapping

MPEG-G stores metadata in the information boxes Dataset Group Metadata (dgmd) and Dataset Metadata (dtmd) [9] using Extensible Markup Language (XML) [10].

To test the compatibility of the GIPAMS architecture with the MPEG-G metadata we have done a mapping of the metadata coming from two public organizations, the European Nucleotide Archive (ENA) [11] and the National Center for Biotechnology Information (NCBI) [12]. The structuring of metadata used in ENA is similar to the one defined in MPEG-G so the mapping is direct, as represented in Table 1. The structuring used at NCBI is more troublesome for us, as the Abstract field does not exist in the NCBI metadata and the Type field is not exactly the same as the one used in MPEG-G. The applied mapping is represented in Table 2.

MPEG-G also provides a mechanism to store additional information in the metadata fields. To test this mechanism we have defined three extensions for the NCBI metadata: one to store an additional identifier to the MPEG-G Sample field, another to store information about the organisms investigated in the project and store multiple investigation centers in a metadata file, and the last one to deal with an additional field in the NCBI BioSample, which contains extra information about the samples that cannot be stored in MPEG-G metadata fields. They are represented in Table 3.

MPEG-G Field	ENA Field
Title	Study - STUDY_TITLE
Type	Study - STUDY_TYPE
Abstract	Study - STUDY_ABSTRACT
ProjectCentre	Study - CENTER_PROJECT_NAME
Description	Study - STUDY_DESCRIPTION
Sample - TaxonId	Assembly - TAXON_ID
Sample - Title	Assembly – TITLE

Table 1. ENA metadata mapping.

MPEG-G Field	NCBI Field
Title	BioProject – Title
Type	BioProject - ProjectTypeSubmission
Abstract	Non existent
ProjectCentre	BioProject – Organization
Description	BioProject – Description
Sample - TaxonId	BioSample - TAXON_ID
Sample - Title	BioSample – TITLE

Table 2. NCBI metadata mapping.

AttributeExtension	NCBI Field
StudyDesign	BioSample - Attribute - study design
BodySite	BioSample - Attribute - body site
AnalyteType	BioSample - Attribute - analyte type
IsTumor	BioSample - Attribute - is tumor

Table 3. AttributeExtension fields.

3.2. File Structure Implementation

As stated in section 1.1, MPEG-G files are structured in hierarchical boxes forming a single file. To avoid having to deal with large files, we have used an alternative approach to simulate this structure, which is using the file system to represent the box structure.

In this approach, every box is represented by a directory, which contains multiple files like the headers, metadata or protection and some subdirectories representing the inner boxes.

The information files still need to be manipulated at bit level, so we have developed a Python script that can generate this whole directory structure and create the information files using valid data. The script can also integrate real metadata and protection policies into the files, using the data provided by the user. The complete MPEG-G file could be constructed from the directories and files stored in disk, if it is needed to share it with some other researcher or organization.

4. Discussion

Having a system like GIPAMS that can deal with genomic information in a modular and secure way may help in providing new services aimed at achieving personalized medicine.

An example of such a service is metadata search. By connecting metadata and genomic data in a more integrated, but flexible and efficient way, researchers may find subjects of interest for their research. The search may also include some information about the genomic data associated to metadata, especially if it can be accessed partially or completely. This point is controlled by means of the access rules defined for genomic metadata and data and of course supported by the encryption/decryption of the information. This could be for example very useful for rare diseases, where a few cases are available and the privacy and security standards should be the highest in order to not to reveal patient identity.

Furthermore, several GIPAMS' implementations could be established at different locations. The possibility of defining global access rules over the metadata stored in each location may facilitate the creation of a federated system providing federated search, still guaranteeing privacy and security. The advantage of having a GIPAMS federation is the fact that we may have several small modular systems, dealing only with their own genomic information (less storage and transmission required), but with the possibility of accessing to metadata describing other genomic studies that may be relevant for them. In a next step, request of the genomic information (or part of it), in addition to metadata search, could be implemented, also in a controlled and secure way.

Finally, the application of different security mechanisms provides privacy protection for the genomic information managed inside GIPAMS. First of all, the communication between the user application and the rest of the modules/services is done through a secure channel. Moreover, tracking of user actions and unique user identification are also implemented. For the protection of the genomic data and metadata, different encryption techniques can be applied. And to control which actions can users perform in the system, privacy protection rules can be defined with a high level of granularity. Although information could be leaked once at the user's application, we should be aware of two relevant facts: 1) users are identified and "trustable", 2) all actions on the information are tracked.

5. Conclusions and Future Work

This paper presents an initial implementation of GIPAMS, a modular architecture for the management of genomic information. Its first implementation is done based on the MPEG-G standard, which organizes genomic information, metadata and protection in a unique structure, facilitating access control, security and efficiency in storage.

The decision of using MPEG-G for the first implementation is twofold: It defines a clear hierarchy for representing and storing different kinds of genomic information and integrates security and protection mechanisms in it since its inception; i.e., by design. Based on this hierarchy, it seems feasible to integrate with other existing genomic information formats, facilitating its search and linkage through all the processes associated to the use of the information.

Moreover, the extension metadata mechanism offered by MPEG-G helps in the inclusion of new metadata, facilitating the implementation of more specific and accurate searches, giving access to more research results.

The implementation of such a system, which defines mechanisms for accessing genomic metadata and data in a secure and controlled way, may help in the implementation of more complex systems, like a federation of GIPAMS.

Apart from improving the federation facilities, next steps include to completely implement the modules conforming GIPAMS, and to extend supported genomic information formats, following a hybrid approach, which might consist on maintaining (at least part of) MPEG-G file structure including metadata and access rules, but supporting other genomic information representation formats in the lower levels.

Acknowledgement

This work is partly supported by the Spanish Government (GenClinLab-Sec, PID2020-114394RB-C31) and by the Generalitat de Catalunya (2017 SGR 1749).

References

[1] genome.gov, The Cost of Sequencing a Human Genome. 2020. https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost
[2] Delgado, J.; Llorente, S. FAIR aspects of a Genomic Information Protection and Management System. European Federation for Medical Informatics (EFMI) – Special Topic Conference (STC) 2021. 2021.
[3] ISO/IEC 23092-1:2020, Genomic information representation - Part 1: Transport and storage of genomic information. 2020. https://www.iso.org/standard/79882.html
[4] Alberti, C. et al. An introduction to MPEG-G, the new ISO standard for genomic information representation, bioRxiv, 2018. https://doi.org/10.1101/426353
[5] Llorente, S.; Rodriguez, E.; Delgado, J.; Torres, V. Standards-based architectures for content management, IEEE multimedia, 2012. https://doi.org/10.1109/MMUL.2012.58
[6] IETF. The OAuth 2.0 Authorization Framework, 2012. https://datatracker.ietf.org/doc/html/rfc6749
[7] IETF. JSON Web Token (JWT), 2015. https://datatracker.ietf.org/doc/html/rfc7519
[8] Keycloak, Open Source Identity and Access Management, 2021. https://www.keycloak.org/
[9] ISO/IEC 23092-3:2020, Genomic information representation - Part 3: Metadata and application programming interfaces (APIs). 2020. https://www.iso.org/standard/75625.html
[10] W3C, Extensible Markup Language (XML) 1.1 (Second Edition), 2006. https://www.w3.org/TR/xml11/
[11] European Nucleotide Archive (ENA). 2021. https://www.ebi.ac.uk/ena/browser/about
[12] National Center for Biotechnology Information (NCBI). 2021. https://www.ncbi.nlm.nih.gov/