The Open Navigation Surface Project

Richard T. Brennan[1,2], Shannon Byrne[3], Brian Calder[2], James D. Case[2,3], David Fabre[4], Barry Gallagher[1], R. Wade Ladner[4], Bill Lamey[5], Friedhelm Moggert[6], Mark Paton[7] and the Open Navigation Surface Working Group

[1] National Ocean Service, National Oceanic and Atmospheric Administration, Silver Spring MD, USA.
[2] Center for Coastal and Ocean Mapping & NOAA/UNH Joint Hydrographic Center, University of New Hampshire, Durham NH, USA.
[3] Science Applications International Corporation, Marine Science and Technology Division, Newport RI, USA.
[4] Naval Oceanographic Office, Stennis Space Center MS, USA.
[5] CARIS Ltd., New Brunswick, Canada.
[6] Seven Cs AG & Co., Hamburg, Germany.
[7] IVS3D Ltd., New Brunswick, Canada.

Abstract

Many hydrographic and oceanographic agencies have moved or are moving towards gridded bathymetric products. Grids provide a robust, powerful and yet simple way to represent geospatial data, but are also much better suited to automatic manipulation than the point-data sets typically used in more traditional hydrographic products (e.g., smooth-sheet selected soundings). However, there is no accepted standard format to allow these grids to be exchanged while still maintaining data and metadata integrity.

The Open Navigation Surface (ONS) Project is an open-source software project designed to provide a freely available, portable source-code library to encapsulate gridded bathymetric surfaces with associated uncertainty values. The data file format, called a Bathymetric Attributed Grid (BAG) contains metadata (e.g., collection platform, authorizing agent, operations carried out on the data, etc.), at least two co-located gridded layers containing the bathymetry and its associated uncertainty, and a list of operator mandated overrides (a.k.a., 'Golden Soundings') indicating changes made to the grid in order to make it 'safe for navigation.' The BAG can also have additional layers of gridded data, or arbitrary data of other forms as optional extensions. A mechanism is provided to digitally 'sign' the BAG, allowing an observer to verify that a person named in the metadata did in fact authorize the contents of the BAG, and that the file has not been modified since it was authorized.

The BAG is implemented using standards-based algorithms and libraries when possible. It uses the HDF5 library to encapsulate the data, ISO19115 for metadata with ISO19139 for XML presentation, and the Digital Signature Standard & Secure Hash Standard for authentication and verification of content. The standard is developed and maintained by the ONS Working Group (ONSWG), managed by an e-mail list. The source code is available via the ONS website. Development is co-operative and voluntary; community participation is encouraged.

Introduction

The Navigation Surface [1, 2] paradigm is a design for a databased alternative to traditional methods of representing bathymetric data. It aims to preserve the highest level of detail in every bathymetric dataset and provide methods for their combination and manipulation to generate multiple products for both hydrographic and non-hydrographic purposes. The advantages of the method over traditional schemes are such that a number of commercial vendors have adopted the technology. However, this means that there is a strong requirement for a method to communicate results in a vendor neutral technology. The Open Navigation Surface (ONS) project was designed to fill this gap by implementing a freely available source-code library to read and write all of the information required for a Navigation Surface.

The Navigation Surface concept requires that in addition to estimation of depth, we must also estimate the uncertainty associated with the depth. In order to make the system suitable to support Safety of Navigation applications, we also require a means to over-ride any automatically constructed depth estimates with 'Hydrographer Privilege', essentially a means to specify directly the depth determined by a human observer as being the most significant in the area (irrespective of any statistical evidence to the contrary). Finally, we must provide data on the data, or metadata, which describes all aspects of the data's life from methods of capture to processing methods, geospatial extents to responsible party. The ONS project provides means to incorporate all of these requirements in a portable, platform neutral, vendor neutral format.

The ONS project has as its primary goal to foster and support the development of a source level library to read and write the data format. A predicate of the project is that the implementation of the format in concrete terms through a generally available source library is more likely to engender acceptance and adoption than a more formal

approach through a standards committee process in a recognized standards authority (e.g., the International Hydrographic Organization). The approach here is to build the object first, and offer it for comment with the full expectation that the first generation product is unlikely to meet all of the goals, and will have to be modified later. Even if there are mistakes in the object structure, however, anyone critical of the design has to suggest an alternative, better, approach to have a valid case. Then, the alternative can be adopted and the whole project can benefit from the suggestion.

This paper describes the design goals of the ONS project, the technology adopted to implement the goals, and the progress made towards a first release of the source code library. Details of the data encapsulation, metadata structure and digital signature schemes are also provided. Finally, we conclude with some perspectives on the future of the project.

Design Considerations

Nomenclature and Basic Structure

The term "Navigation Surface" was coined to describe the combination of a data object representing the bathymetry and associated uncertainty, and the methods by which such objects could be manipulated, combined and used for a number of tasks, including products in support of safety of navigation. These multiple goals have led to some confusion over what exactly constitutes a Navigation Surface. To avoid any further confusion, a revised nomenclature has been designed.

In the ONS model, a unit of bathymetry is termed a Bathymetric Attributed Grid (BAG). A single BAG object represents one contiguous area of the skin of the Earth at a single resolution, but can represent data at any stage of the process from raw grid to final product. The name Navigation Surface (NS) is reserved for a final product BAG destined specifically for safety of navigation purposes. The status of any particular BAG is distinguished solely by the certification section of meta-data embedded in the file.

By definition, all components of a BAG use only metric SI units, and can be in projected or unprojected units. In unprojected (geographic) grids, decimal degrees are used for all coordinates; in projected grids only meters are used. A limited number of horizontal datums are supported. All parameters necessary to fully define the horizontal datum are contained within the BAG. A limited number of projection coordinate systems are supported. For projected grids, all parameters necessary to convert units between projected X, Y and geographic are saved within the BAG. A right-handed coordinate system is used, and hence the *z* component is increasing to shallower areas (i.e., positive up); the term "elevation" is therefore used in preference to "depth". Negative elevations are therefore below datum, positive above; a limited number of vertical datums are defined as typically used in hydrographic work. Grids are stored in row-major order, with geo-referencing defining the south-west corner of the grid. The value in the first element of the grid should be associated with the geo-referencing position as a point, rather than representing any areal extent. Times are always represented in UTC and seconds since 1970-01-01/00:00:00.

Standards Compliance

Whenever possible, the BAG object will be defined through established standards. A number of groups have considered the problem of structured data description, meta-data and digital security. ONS will adopt these as they make sense. However, it is not the intention of the ONS, at least initially, to attempt to fully define a standard for gridded data. The sole intent is to provide a working prototype and associated documentation.

Data Encapsulation

BAG objects are intended to be system independent and persistent. Endian issues are addressed at the level of the data encapsulation.

Metadata

Metadata is a mandatory part of the BAG format. The BAG format defines metadata as all auxiliary information required to interpret the basic data itself. This includes not only the details of the data collection (the who, what, where, why and when), but also the processing stages applied to the data since capture and legal statements about the validity, intended used, or expected disposition of the data in the BAG. The metadata will be embedded in the BAG file itself, rather than being a separate file. This is intended to ensure that the metadata and the data cannot be separated.

Security and Authenticity

Since the final intent for BAG objects is as legal archives for hydrographic data, and products to support safety of navigation, it is essential that there is a way to prove that a BAG has been authorized by someone with appropriate credentials, and that it has not been modified (either intentionally or through transmission errors) since it was so

certified. Both of these goals can be achieved through an appropriate cryptographic hash function and digital signature scheme. In the BAG format, the authentication method will certify the whole of the file, and provide a method to link the authorization authority with a particular person or entity. The security scheme is not intended to prevent use of the data – no encryption of the data is implied – and there is no direct implication of what is being certified in the BAG. It is a requirement of the format that before the BAG is digitally signed to certify the contents, an element is added to the metadata indicating the intent of the signature. This might be as simple as 'data is complete and verified' to 'suitable for compilation', 'suitable for database insertion', or even 'certified for navigational use'. Presence of a signature and certification in a BAG does not imply that the signing entity has the appropriate level of authority to make the attached certification. Verification of a signing entity's credentials is auxiliary to the BAG structure, and is not considered. Implementation of a certificate scheme, key management and other details of a public-key infrastructure required to support a digital signature scheme are also not defined in the BAG structure. These are considered a matter for the adopting agency.

Mandatory Elements and Extensibility

To be considered valid, a BAG must contain a metadata element, an elevation grid, an uncertainty grid and a change-list indicating any modifications made to the grid due to hydrographic concerns. Each of the elements (except the metadata) can have element specific metadata (e.g., minimum and maximum elevation in the grid). The meaning of 'uncertainty' has not been defined directly in the baseline model, since it may be different for different users, and may change through the lifetime of the BAG from raw data to final product. (For example, the uncertainty at a base BAG contains information about the raw data processed, and is intended for use by hydrographers and cartographers in making compilations; uncertainty in a product BAG is intended to indicate reliability of the depths for the end user, and may be significantly different.)

The change-list element can be empty. Otherwise, it contains the original depth and uncertainty values for every node that has been modified in the BAG due to operator intervention. This is a required feature to ensure hydrographer override and hence safety of navigation. To ensure that the base BAG product is safe, the values in the elevation grid are the modified (safe) values; since these are read unless the user specifically requests the original value, the BAG is fail-safe.

The BAG's internal structure allows for extension components to be defined at the same level of the file as the mandatory elements. Extensions will be considered as they are proposed by users. The only restriction is that the full structure of the proposed extension must be provided before adoption into the format definition.

Support and Administration

The BAG format will be defined through its format document and implemented via a platform-independent source-code library that is generally available from a CVS server. Background information and documentation will be provided via a web-site at `http://www.openns.org`. Availability of the source code for the access library is intended to facilitate community acceptance of a single implementation of the BAG format, thereby maximizing interoperability between and amongst various user groups. The access library will be implemented using the ANSI standard C programming language.

The ONS is a community-led effort, and the BAG format is supported by the volunteer support of the ONSWG. Once the first release of the source-code library is complete, the intent is to monitor and support the format through an Architecture Board (ONSAB) co-opted from the members of the ONSWG and potentially others as time progresses. The ONSAB will co-ordinate bug-reports, change requests and feature improvements in the BAG format, and accept requests for and definitions of new sub-objects.

Technology

Data Encapsulation

To support the requirements for platform independence, large file support (> 2 Gigabytes), and multi-component objects, a suitable data encapsulation layer is required. This layer translates internal data into an external representation and provides the logical hierarchical structure for the components of the BAG. Hierarchical storage of scientific data is a common problem for which standard solutions exist. ONS has adopted the Hierarchical Data Format V (HDF-V) [3], supported by the National Center for Supercomputing Applications. HDF-V is widely used is scientific applications for storage and manipulation of very large datasets, is implemented via a platform independent library available in source code form, and is supported by a number of tools for visualization and manipulation. The logical BAG structure is shown in Figure 1; HDF-V maps this as a hierarchical structure in a model reminiscent of a file structure, Figure 2.

Meta-data Encapsulation

Models for geo-spatial metadata are now common. Both FGDC and ISO have addressed the issue. The BAG structure will adopt the ISO19115 standard for geo-spatial metadata [4], and the ISO19139 encoding of this [5], which uses XML to structure the metadata. An abstract of the XML schema for BAGs is shown in Figure 3, and an example meta-data file is shown in Figure 4. FGDC are defining a cross-walk for ISO meta-data schemes [6], which allows for translation between the two formats if required.

Signatures and Authentication

Schemes to provide an entity with the same properties as a physical signature have been developed primarily for use in legal transactions conducted via electronic means. Based on algorithms for public-key encryption [7], a Digital Signature Algorithm (DSA) computes a number from the file to be signed and combines it with some information known only to the signer (the private or secret key) in order to make the signature (a very large number). This signature has the property that it is very difficult (essentially impossible) to compute without knowledge of the secret key. However, if you have the complement of the signer's secret key (known as the public key), then it is easy to verify that the person who affixed the signature to the file did in fact have knowledge of the secret key. The signature and file are inextricably linked: if the signature is modified, it will not verify; if the file is modified, it will not verify. Hence, the DSA also provides protection from errors due to transmission and ensures that a grid prepared for safety of navigation purposes cannot be falsified without detection.

The ONS DSA uses the US Federal Information Processing Standard (FIPS) Digital Signature Standard (DSS), FIPS 186-2 [8], which uses the Secure Hash Standard (SHS), FIPS 180-2 [9] as a sub-component. There is no standard implementation of either DSS or SHS, and the US National Institute of Standards and Technology (NIST) will only certify compiled executables, and not source code. To implement the encryption primitives required for the algorithm and maintain cross-platform source code support, the BeeCrypt library [10] was used.

Implementation

Base Libraries

The ONS access library is based on libraries for HDF-V access, compression, XML parsing, and encryption and signatures. The data encapsulation solution is based on HDF-5 version 5.1.6.1, available from the National Center for Supercomputing Applications in both source code form and in binary form for several platforms of interest. The decision to adopt HDF-5 was made after several alternative solutions were considered. Features considered important in the evaluation process included: open unrestricted nature of target format, file I/O speed, platform independence, support for large datasets and large file sizes, currently existing capability, and long-term software support. While the initial release of the BAG format and access library will not take full advantage of all of the features currently available in HDF-5, adoption of HDF-5 as the encapsulation solution provides a solid foundation for the current needs of the ONS and expected directions of future growth.

HDF-5 depends on the compression services provided by zlib version 1.1.4 and szip 1.1. As with HDF-5, these two packages are available as both binary and source code distributions.

In order to ensure consistency in the transformation of positional data between geographic and projected X,Y, the ONSWG decided to adopt the *geotrans* package available from the US National Geospatial Intelligence Agency. *Geograns* provides the algorithms for converting positional data between geographic to projected X,Y for a wide variety of projection definitions. *Geotrans* also provides the algorithms for converting geographic data between various horizontal datums. *Geotrans* version 2.2.5 is the current version of as July 2004. More information on *geotrans* can be found at: http://earth-info.nga.mil/GandG/geotrans/.

The encryption and signature structures are based on BeeCrypt 3.1.0, which is freely available, supported, and available in source-code form which compiles cleanly on the ONS target platforms. Other alternatives were tested, including GCrypt and Crypto++, but these proved to be problematic for the target application. GCrypt theoretically compiles on Win32 platforms, but requires a cross-compilation system to build and the process is convoluted. Crypto++ is heavily C++ oriented and would thereby limit integration of the library with other language bindings. No special procedures were required to build BeeCrypt, and a Microsoft Visual Studio 6 sub-project is available. Some limitations of BeeCrypt were identified, including the requirement that the SHA-1 algorithm be used for constructing message digests. While this is technically by the letter of FIPS 186-2, many experts now agree that the key-range of SHA-1 (160 bits) is sufficiently limited to cause the potential for hash-collision (i.e., one BAG file having the same digest as another, and hence causing a limitation of the strength of the protection implied by the signature scheme), and are recommending use of SHA-256 or SHA-512 (256-bit or 512-bit digests) instead. An extension of BeeCrypt to support this may be possible in the future.

Interfaces

The ONS access library provides a set of generic data type definitions and application program interfaces (APIs) intended to isolate the application developer from the details of the underlying HDF, XML, and cryptographic implementations. The data type definitions and APIs exported from the access library follow all of the BAG design guidelines and definitions. Data types are defined with the right handed coordinate system and metric SI units. The initial software release will provide support for projected grids with nodes regularly spaced on a flat, horizontal surface, and geographic grids with nodes regularly spaced in units of decimal degrees on the ellipsoid. The access library APIs will provide services for creating a new BAG, or accessing and existing BAG. Services will be provided for reading and writing individual nodes, and for reading and writing by grid row. The initial release of the access library will provide the basic structural definitions and functions necessary to operate on BAGs at a fundamental level. The approach taken here is to provide the basic mechanisms necessary to access the data in a structured way via simple building blocks that will allow for straightforward future growth of library capability.

Meta-data Schemas and Interfaces

The meta-data schema employed by the BAG format is a *profile* of the ISO19115 specification (Figure 3). The *profile* was comprised by the ONS to cover the core requirements of the specification, plus add the additional fields necessary to fully describe the gridded data stored within the BAG. The profile can be broken down into two logical categories: *informational* and *processing*. The informational components describe details on the origin of the data, producing organization (i.e. owner/creator), contact information, date of creation, etc. The processing components are used to describe the physical characteristics of the gridded data and include geographic extent, grid resolution (cell spacing), horizontal and vertical coordinate system descriptions, and the procedures and processes used to create the gridded surface.

To implement ISO19115 standard, the ISO19139 XML Meta-data schema was chosen. XML allows the meta-data to be stored and retrieved in a system independent format, and provides a machine and human readable version of the meta-data at no extra cost. The data can be viewed and edited in a simple text editor, or more sophisticated displays can be created by using any XML data parser, many of which are freely available. The master copy of the schema will be held at a publicly available URI associated with the ONS website.

To keep the BAG API simple and flexible, and to avoid exposing any third party XML parsing library, the interface for retrieving or providing the XML meta-data will be done using simple UTF-8 character streams. Internally, the XML data will be verified using the ISO19139 meta-data schema. This ensures that the meta-data for any BAG is complete, and thus can be read by any user of the ONS libraries, but grants flexibility to the end user by not exposing a particular XML engine.

Security Infrastructure

The ONS/DSA ensures that the file has not been modified and that the public key used matches the private key used to sign the file. However, it does not provide for construction, distribution or security of key pairs, or certification that the person claiming to own the secret key used does in fact own that key. Although these details are strictly outside the domain of the ONS, they are critical to a workable scheme. In addition to the `libonscrypt` module used to provide basic services, the `libexcert` module is provided to implement a simple certificate scheme.

Distribution and security of key pairs is implemented using an auxiliary hardware token, implemented in the example using a standard software licensing USB dongle [11]. On construction, the owner of the key pair must be physically present and prepared to provide a pass-phrase of 20-50 characters. As soon as the key pair is computed, the secret key is encrypted using the Advanced Encryption Standard (AES), FIPS 197 [12]; the (symmetric) key is computed from the pass-phrase using SHS. The encrypted secret key is then written into the on-board memory of the USB dongle, Figure 5. The dongle is designed to be tamper-proof and contains its own encryption technology to avoid having the data read without appropriate software. The dongle is protected by software password numbers so that it cannot be read or re-programmed without the software designed for the task, avoiding end-around attacks. In conjunction with the AES encryption, the probable time to recover a secret key from the USB dongle is significantly higher than the time required to notice that a USB dongle has been compromised (e.g., lost, stolen) and repudiate the associated public key. (The public key is written into a separate plain-text or XML file, known as a certificate, which can be freely distributed.)

To sign a file, the signer provides the USB dongle, and the associated pass-phrase. The software module re-computes the AES key, reads and decodes the secret key from the dongle, and uses it to compute the signature in what appears to the user to be an atomic operation (Figure 6). After the signature is computed, the dongle is removed and secured. To verify a file's signature, the public key certificate is obtained, and the software module simply extracts the appropriate information (Figure 7).

Certification of the identity of the owner of a key pair is done by having another authority sign the public key certificate, augmented to include identification information (e.g., the owner's name). Signature occurs as above, save that the Certificate Signing Authority's (CSA) USB dongle is used to construct the signature (Figure 8). The CSA's public certificate is used to verify a user certificate before use (Figure 9); the CSA's public certificate can be signed by another authority to provide higher levels of authentication, or may be self-signed to terminate the chain of authentication. Since the veracity of certificates relies on the CSA's key pair security, it follows that the CSA must be trusted by all participants in the scheme, and must have a secure method of constructing its key pairs. Consequently, the CSA would typically be a government agency.

Outreach, Support, and Administration

The ONS project has been implemented, developed and coordinated primarily through a pair of e-mail lists: `navsurf_general@ccom.unh.edu` (for general administrative, policy and organizational traffic to a wider audience), and `navsurf_dev@ccom.unh.edu` (for detailed technical issues and development traffic between active developers of the library), both hosted as a service to the community at the Center for Coastal and Ocean Mapping & NOAA-UNH Joint Hydrographic Center (CCOM-JHC) at the University of New Hampshire. Membership of the mailing lists is un-restricted and requests to be included should be directed to `navsurf_join@ccom.unh.edu`. The current mailing lists contain members from US (NAVO, NOAA, NGA, and NGDC), Canadian, British and Australian hydrographic agencies and organizations, sonar equipment manufacturers, hydrographic software manufacturers, electronic mapping software manufacturers, and academia. A basic web-site containing information on the project will be available as `www.openns.org`. Initially, the web-site will be used as a placeholder to convey information on the project to the community. However, as the project progresses, the web-site will act as a release notification system, a location for binary release downloads, and a clearing-house for bug reports, enhancement requests and implementation information. Choice of the ONS Architecture Board will be delayed until the first full release of the source code.

The source code for the project is hosted on a Concurrent Versioning System (CVS) repository server, `cvs.ccom.unh.edu`, also hosted at CCOM-JHC. The source-code for the project is available, and requests to access the software should be directed to `navsurf_access@ccom.unh.edu`. At present, the source-code is a 'work-in-progress' on its way to a formal beta release. The base library structures are complete, as are the XML schema and library, and the cryptographic libraries. A generalized build system for Win32 platforms has also been implemented, and Win32 is the primary development and test environment for the source. A `makefile`-based build system for Unix-style platforms is under construction.

Summary

The Open Navigation Surface Project was established to design and develop a specification for hydrographic gridded data, and to implement a freely available source code library to support the specification.

An ONS object is known as a Bathymetric Attributed Grid (BAG), and consists of meta-data, a grid of elevation data, a corresponding grid of uncertainties describing the elevation data, and a list of modifications made to the grid by a hydrographer in support of safety of navigation. Extra information may also be present in the file if required. After the body of the data, a digital security system block is appended to provide verification and authentication services for the whole of the file.

The project is supported by the community for the community and it is intended that the source code for the project will be available to all interested parties. Community participation in the project is positively encouraged.

References

[1] Smith, S. M. (2003). *The Navigation Surface: A Multipurpose Bathymetric Database*, Masters Thesis, Center for Coastal and Ocean Mapping & Joint Hydrographic Center, University of New Hampshire.

[2] Smith, S. M., Alexander, L., & Armstrong, A. A. (2002). 'The Navigation Surface: A New Database Approach to Creating Multiple Products from High-Density Surveys', *Int. Hydro. Review*, Vol. **3**, No. 2, pp. 12-26.

[3] See the HDF-V support web-site, `http://hdf.ncsa.uiuc.edu`.

[4] ISO Standard 19115:2003 Geographic information – Meta-data (`www.iso.ch`)

[5] ISO Proposed Standard 19139: Meta-data – Implementation Specification (`www.iso.ch`)

[6] Pearsall, R. A., (2001) FGDC Profile(s) of the ISO Technical Committee 211 Metadata Standard (ISO 19115) (`http://www.fgdc.gov/standards/documents/proposals/iso_metadata.pdf`)

[7] Schneier, B. Applied Cryptography: protocols, algorithms, and source code in C. Wiley (New York), 1996.

[8] The Digital Signature Standard. National Institute of Standards and Technology, FIPS 186-2, 2000 (http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf)

[9] The Secure Hash Standard. National Institute of Standards and Technology, FIPS 180-2, 2002 (http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf)

[10] Deblier, R. (2004) BeeCrypt API Documentation, http://users.telnet.be/robert.deblier/beecrypt.

[11] Aladdin Knowledge Systems, HASP English Programmer's Guide, 2003, (http://ftp.ealaddin.com/pub/hasp/new_releases/docs/HASP_Manual_EN.zip)

[12] The Advanced Encryption Standard. National Institute of Standards and Technology, FIPS 197, 2001 (http://csrc.nist.gov/publications/fips/fips197/fips197.pdf)

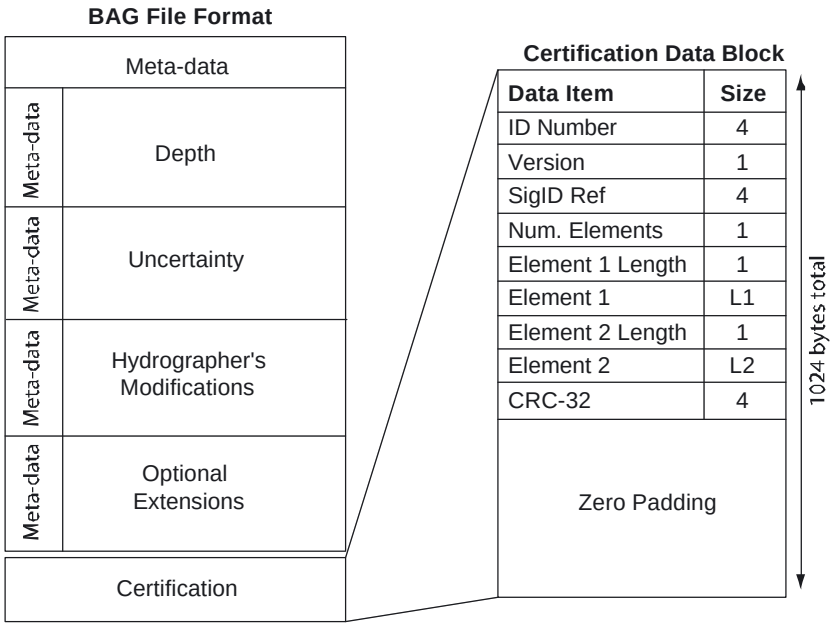

Figure 1: Logical structure of the BAG file format and digital signature certification segment. Layer-specific meta-data is limited in nature, for example numerical bounds on the data; file-specific meta-data should contain all information required to represent the file, including capture parameters, processing specification and certification of the data.

```
Group “/” {
        Group “BAG Version” {
                Attribute “Version”{
                        DATATYPE
                        DATASPACE
                        DATA {}
                }
        }
        Group “BAG #” {
                Attribute “Id Name” {
                        DATATYPE
                        DATASPACE
                        DATA {}
                }
                Attribute “metadata” {
                        DATATYPE
                        DATASPACE
                        DATA {“XML…”}
                }
                Group “surface” {
                        Dataset “elevation” {
                                DATATYPE
                                DATASPACE
                                DATA {}
                                Attribute “metadata”{
                                        DATATYPE
                                        DATASPACE
                                        DATA {min,max,precision}
                                }
                        }
                        Dataset “uncertainty” {
                                DATATYPE
                                DATASPACE
                                DATA {}
                                Attribute “metadata”{
                                        DATATYPE
                                        DATASPACE
                                        DATA {min,max,precision}
                                }
                        }
                        Dataset “tracking list” {
                                DATATYPE
                                DATASPACE
                                DATA{}
                                Attribute “metadata”{
                                        DATATYPE
                                        DATASPACE
                                        DATA{number of changes}
                                }
                }
                Group “extension” {
                }

        }
}
```

Figure 2: HDF-V structure of the BAG file format. The layers of the BAG are mapped into hierarchical elements with data types specified. Certification is carried out external to the HDF-V structure so that all of the file can be protected by the same signature structure. Currently, only one BAG grid is supported within the file, although multi-BAG HDF-V files might be a future extension.

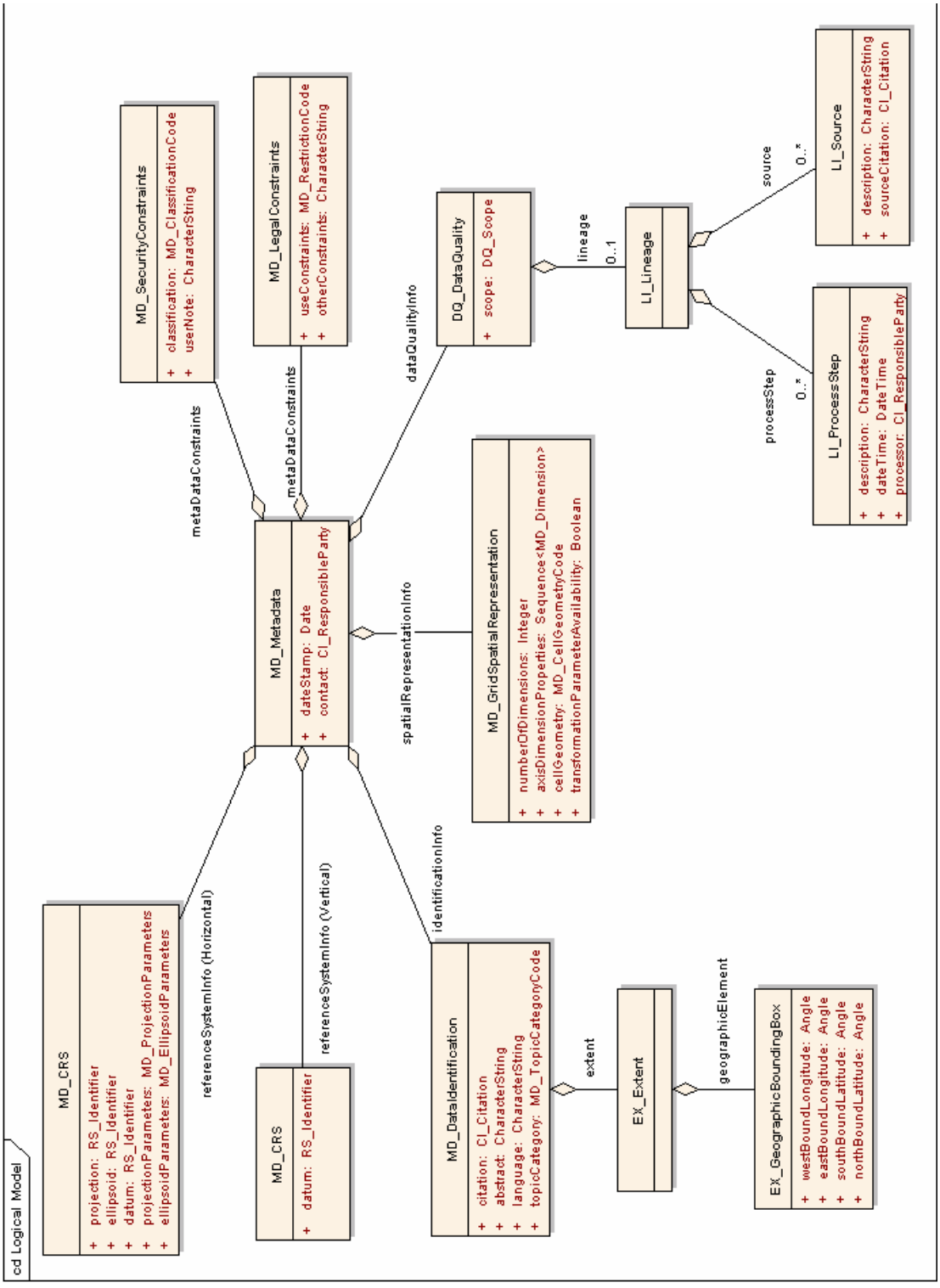

Figure 3: XML schema for ISO19139-style encoding of ISO19115 meta-data.

```
<?xml version="1.0" encoding="UTF-8"?>
<MD_Metadata xmlns=http://metadata.dgiwg.org/smXML
             xmlns:gml="http://www.opengis.net/gml" xmlns:xlink=http://www.w3.org/1999/xlink
             xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
             xsi:schemaLocation="http://metadata.dgiwg.org/smXML
             C:\ISO\smXML\metadataEntity.xsd">
     <identificationInfo>
         <MD_DataIdentification>
             <citation>
                 <CI_Citation>
                     <title>Portsmouth Harbor</title>
                     <date>
                         <CI_Date>
                             <date>2001-11-25</date>
                             <dateType>publication</dateType>
                         </CI_Date>
                     </date>
                     <citedResponsibleParty>
                         <CI_ResponsibleParty>
                             <individualName>CDR. J. Q. Surveyor</individualName>
                             <positionName>Chief of Party</positionName>
                             <role>hydrographerInCharge</role>
                         </CI_ResponsibleParty>
                     </citedResponsibleParty>
                 </CI_Citation>
             </citation>
             <abstract>
                 This BAG is the result of a primary hydrographic survey of Portsmouth Harbor,
                 NH.  The survey was also source data for the Second International Conference
                 on Shallow Water Survey, Portsmouth, NH 2001.
             </abstract>
             <language>English</language>
             <topicCategory>Elevation - Bathymetry</topicCategory>
             <extent>
                 <EX_Extent>
                     <geographicElement>
                         <EX_GeographicBoundingBox>
                             <extentTypeCode>true</extentTypeCode>
                             <westBoundLongitude>-75.0</westBoundLongitude>
                             <eastBoundLongitude>-74.0</eastBoundLongitude>
                             <southBoundLatitude>46.5</southBoundLatitude>
                             <northBoundLatitude>47.5</northBoundLatitude>
                         </EX_GeographicBoundingBox>
                     </geographicElement>
                 </EX_Extent>
             </extent>
         </MD_DataIdentification>
     </identificationInfo>
     <metadataConstraints>
         <MD_LegalConstraints>
             <useConstraints>otherRestrictions</useConstraints>
             <otherConstraints>Not for Navigation</otherConstraints>
         </MD_LegalConstraints>
     </metadataConstraints>
     ...
     <contact>
         <CI_ResponsibleParty>
             <individualName>CAPT. S. B. Collins</individualName>
             <role>chiefOfDivision</role>
         </CI_ResponsibleParty>
     </contact>
     <dateStamp>2001-11-28</dateStamp>
</MD_Metadata>
```

Figure 4: Example XML encoded meta-data extract. The meta-data is human-readable and editable in any text-editor (although specialist XML editors also exist), and contains information on data capture, processing, spatial extents, intended use, responsible party, etc. The XML string is simply encapsulated in the HDF-V structure.

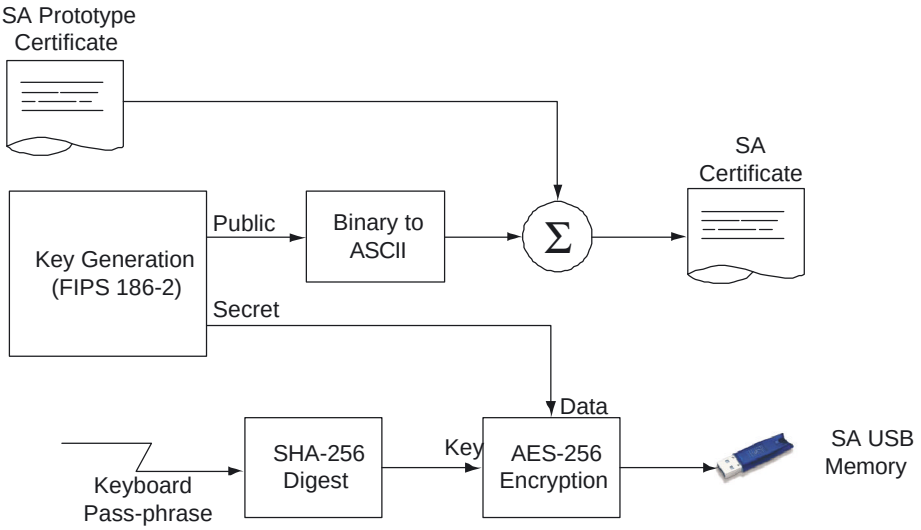

Figure 5: Construction and security scheme for Signature Authority (SA) key pairs. A prototype certificate with the appropriate information on the key owner is augmented with the public part of the key pair, while the private (secret) part of the key pair is encrypted with a symmetric algorithm (AES-256) using a key constructed from the key owner's pass phrase. The encrypted key is then written to a USB hardware dongle, which has a number of security features to avoid unauthorized tampering.

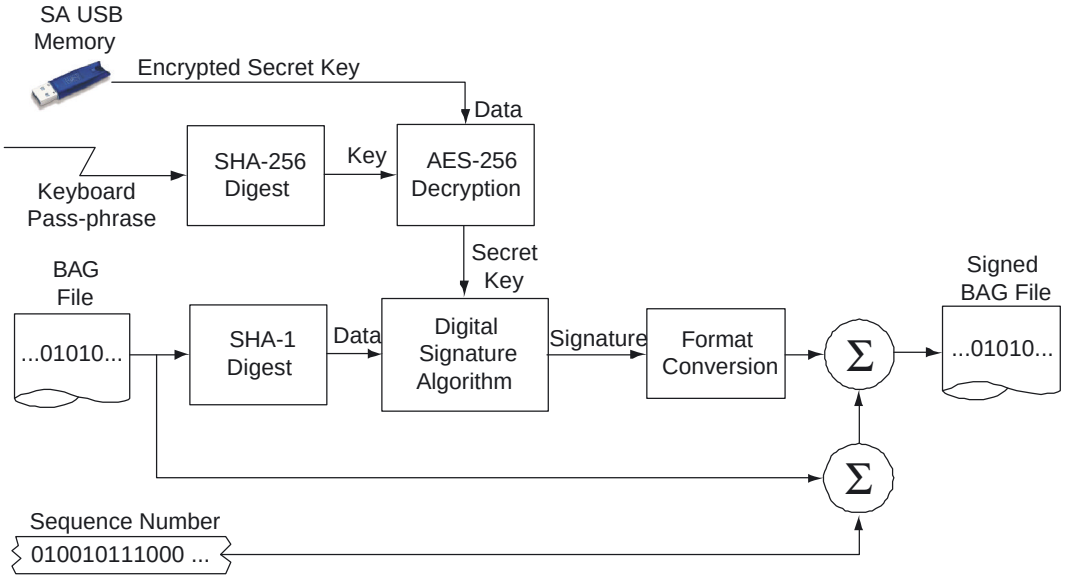

Figure 6: Signature of a BAG file. The Signature Authority (SA) provides the pass-phrase appropriate to the key. The phrase is used to reconstruct the symmetric key for AES-256, and the secret is decoded for use in the signature process. The digital signature is combined with the BAG file and a sequence number (used to identify which statement in the BAG meta-data the signature authenticates) to form the signed BAG.

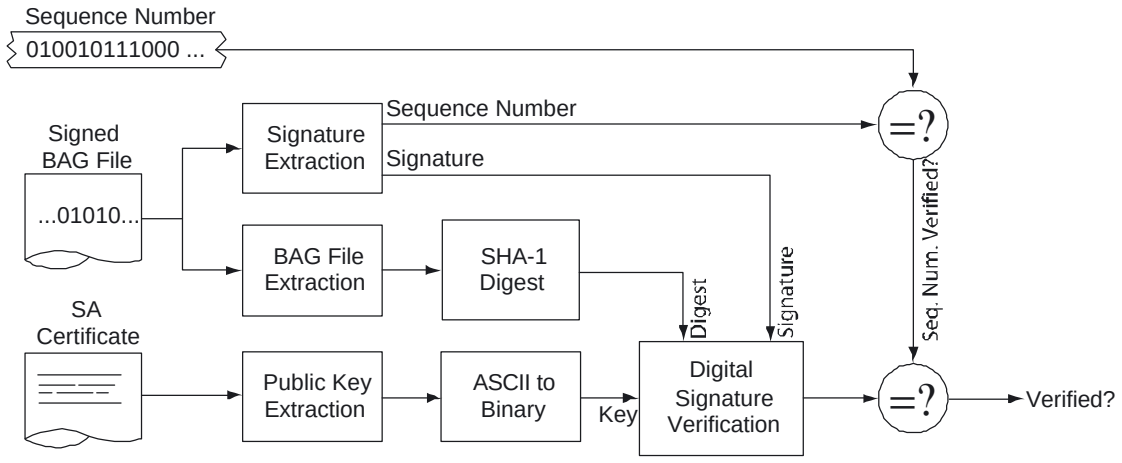

Figure 7: Verification of the signature on a BAG file. The Signature Authority's (SA) certificate is read for its public key component. The signature of the BAG file is extracted, and the summary of the BAG used to construct the signature (the SHA-1 digest) is re-computed. The signature algorithm's verification process is applied and the result is compared with the stored signature. The user-supplied sequence number is also checked to ensure the correct meta-data certification is being considered.

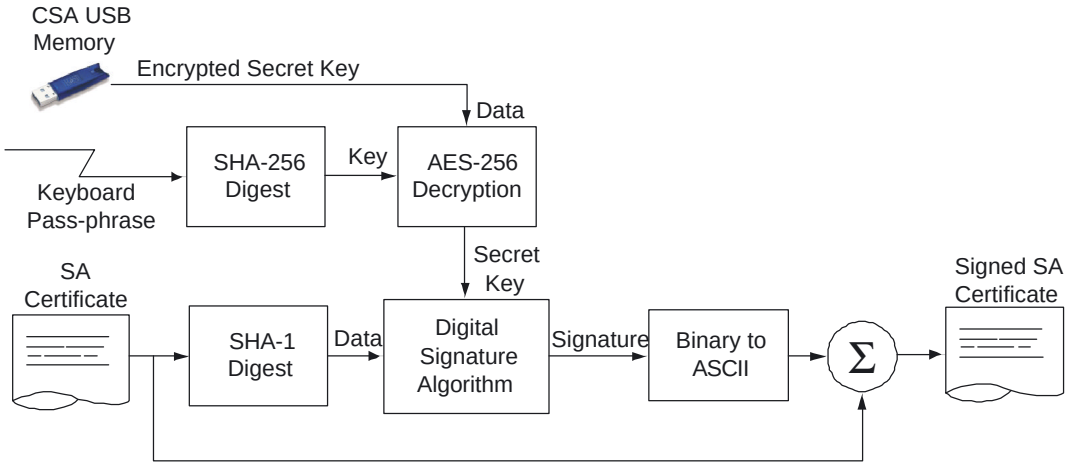

Figure 8: Signature Authority (SA) certificate signature using the Certificate Signing Authority's (CSA) signature dongle. Signature of the certificate matches the scheme for signature of the BAG file, except that the CSA's key is used and the signature is stored in ASCII rather than binary mode.

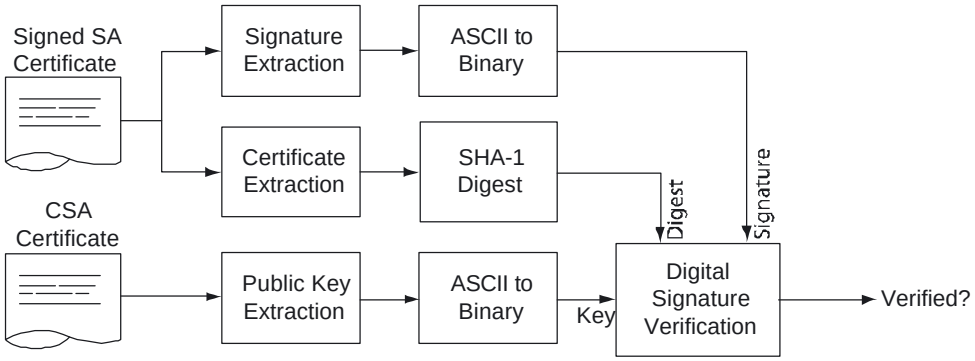

Figure 9: Verification of a Signature Authority (SA) certificate. The process mirrors the BAG verification process except for data formats.