Journal of Information Processing Systems, Vol.6, No.3, September 2010 DOI : 10.3745/JIPS.2010.6.3.359

Access Control to Objects and their Description in the Future Network of Information

Éric Renault*, Ahmad Ahmad* and Mohamed Abid*

Abstract—The Future Internet that includes Real World Objects and the Internet of Things together with the more classic web pages will move communications from a node-centric organization to an information-centric network allowing new a paradigm to take place. The 4WARD project initiated some works on the Future Internet. One of them is the creation of a Network of Information designed to enable more powerful semantic searches. In this paper, we propose a security solution for a model of information based on a semantic description and search of objects. The proposed solution takes into account both the access and the management of both objects and their descriptions.

Keywords—Future Internet, Network of Information, Security, Storage Space, Access Rights

1. Introduction

In today's networking environment, everything is mobile, dynamic, and virtually connected to everything else. This evolution and the ensuing explosion in communications and interactions between objects challenge current host-centric network architectures. Scaling to billions of dynamically and opportunistically interacting objects requires new information object modeling and description, new naming and security mechanisms. The concept of the host having the most significant importance is hopefully changing. Networking is evolving towards a generation where Internet communications move from point-to-point conversations between hosts to point-to-multiparty or multiparty-to-multiparty information dissemination. This generation could be specified as the third generation Internet including the information-centric Network of Information (NetInf) defined in [1]. This kind of network has to deliver dissemination and non-dissemination of objects, reducing unwanted traffic and connecting the digital with the physical world. The goal of information-centric networks is to make all available information easily accessible to the user. These networks rely on the concepts of information and data objects which make the role of hosts less significant compared to their conventional counterparts. The users are interested in the objects themselves and hardly ever on their locations. In other words, the information, not the delivering party, is put in the foreground. This network definition is the main goal of the 4WARD project [2, 4], that

※ This work has been supported by the IST 7th Framework Programme Integrated Project 4WARD, which is partially funded by the Commission of the European Union. The views expressed in this paper are solely those of the authors and do not necessarily represent the views of Institut Télécom — Télécom SudParis or the respective projects and sponsors. The authors would like to thank Mr Djamal Zeghlache for his valuable contribution and support.
Manuscript received April 5, 2010; first revision August 18, 2010; accepted August 24, 2010.

Corresponding Author: Éric Renault
* Dept. of Wireless Networks and Multimedia Services, Institut Télécom - Télécom SudParis, Évry, France (eric.renault@it-sudparis.eu, ahmad.ahmad@it-sudparis.eu, mohamed.abid@it-sudparis.eu)

refers to this type of networking architecture as *NetInf* [1].

NetInf is communication architecture for the Networking of Information based on the information-centric communication paradigm. By adopting an information-centric model, NetInf helps end-users manage information differently; users are able to control the changes in the information values and descriptions, classify the information assets into logical groups and make them available from anywhere. NetInf deals with the dynamic nature of information, tracking their changes from creation to deletion or archiving.

Many problems have been foreseen for the Future Internet [5-7] with consideration of security issues [8, 9] being limited to a scope of publish/subscribe approaches. Our goal is to address security for information networking at the global level with a focus on embedding security mechanisms in the information-centric paradigm of NetInf. Motivated by this issue and by the need of linking the information-centric approach to information security, this paper defines security considerations for NetInf and proposes an efficient security solution which can overcome hurdles associated with information management. Based on the nature of NetInf and its components, access control is seen as one dominant and necessary factor for such a network. The proposed architecture is hopefully not limited to this facet. As NetInf differentiates between BOs (Bit-level Objects) and IOs (Information Objects) to store respectively the content of objects themselves and the sets of information to manage and semantically describe objects, this article presents how IOs could be used to ensure that only granted users can access the content and/or the description of objects, whatever their type (digital objects, real objects, etc.).

The rest of the paper is organized as follows: hypotheses upon which our contribution is based are presented in Sec. 2. Sec. 3 describes the architecture of the Network of Information and Sec. 4 presents the inclusion of access rights and management in NetInf. An evaluation is given in Sec. 5. Finally we challenge our own hypotheses to determine if our solutions hold and assess the potential impact on our proposals in Sec. 6.

2. HYPOTHESES

Our design of a Network of Information for the Future Internet as presented below is based on a small set of most stringent and realistic hypotheses. Considering that the Future Internet may deviate from our assumptions, the last part of the paper is devoted to an analysis of the consequences of our approach if different hypotheses are applied.

No trusted third party The current design of NetInf as described in [10] includes three main parties: a storage space to store objects, a dictionary used to store and manage metadata and provide an efficient way to perform searches, and end-users that can be real people or any kind of processes or user agents. In order to ensure the security of NetInf, it is highly desirable that no other party be involved so as to limit the possibilities of attacks.

User identity not stored in the Network of Information One of the main characteristics of the current Internet is its ability to provide an open and free communication space. For the Future Internet, it is also highly desirable that people are able to express themselves freely. Especially, allowing anonymity is very important even though any anonymous user should be able to claim her/his authorship after revealing her/his identity.

Reliability of the Network of Information, the storage system and the routing For the description of the proposed solution, all these key elements are supposed to be reliable, i.e. they can neither be corrupted nor subject to any malfunction. This hypothesis is clearly the strongest one and therefore will be the most challenged one at the end of the paper.

3. Network of Information Models

The work presented here has been built based on two complementary models: on the one hand, the object model aims at specifying the different kinds of objects managed by the Network of Information; on the other hand, the information model aims at allowing the description of objects provided by end-users and applications. These two models are just provided for reference to the reader as more information is available at [1] and [11] for the object model and the information model respectively.

3.1 Object model

The Network of Information is able to manage two different types of objects: Information Objects on the one hand and Bit-level Objects on the other hand. Fig. 1 shows the relationship between the objects.

A Bit-level Object (BO) is the digital representation of the associated object. From the Network-of-Information point-of-view, this is just a set of bits the meaning of which is only relevant to the end-user and/or application. From the user perspective, a Bit-level Object is typically a matter of content; like a web page, a sound track, a movie, etc. It can also be a means to access non-digital objects like mobile phones or RFID tags or any other relevant device which may serve of interest to be connected to via the Future Internet. In this case, the Bit-level Object can be a process the user can use to access the associated non-digital object.

An Information Object (IO) contains the information related to the objects that are not part of the content. These sorts of information may be of two different kinds:

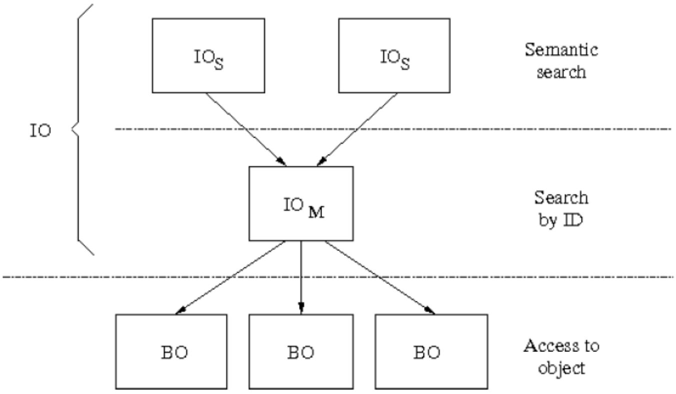

Fig. 1. The object model

- *Semantic Information Objects* (IOs) stores the semantic description of an object, i.e. all the information the owner of an object considers useful to understand and/or search for the object.
- *Management Information Objects* (IOm) stores the information associated to the object that is relevant for its management from the NetInf point of view.

If Semantic Information Objects are managed by users, Management Information Objects are managed by NetInf according to the information provided by their owners.

Some digital objects may grow to a very large size (this is typically the case of videos). As a result, more than one BO can be associated with an object. This approach is very similar to the concept of chunks as used in BitTorrent [12]. In the same way, more than one Information Object can be associated to with a given set of BOs. There can be at the same time an IO for the semantic description of the object and another one for its management. Moreover, there can be concurrent semantic descriptions for a single object, for example if two descriptions are provided for two different kinds of users with two distinct access rights and/or points of view.

A connection to the current Internet can be made to understand BOs and IOs, and highlight the interest to separate them. A web page available on the Internet is mainly composed of the three following elements: the URI that indicates the location of the page, the content that is to be interpreted by the web browser and a set of information used to describe the content of the page (the meta tags of the HTML language), this last element not being mandatory. In the scope of the Object model, the content of the web page is BO, the URI is the part of the IOm and the meta tags are stored inside an IOs.

3.2 Information model

Together with the object model as presented above, an Information model has been developed. The Information model is in charge of defining how Information Objects shall be presented and stored in NetInf. The one developed in the scope of the Network of Information is called the Metalist Model, as it aims at including lists of metadata. The Metalist model includes a set of basic functionalities that allows an easy-to-use management of IOs, whenever it is a Semantic or a Management IO. Three ways have been identified to specify metadata in an Information Object using the Metadata Model (see Fig. 2):

- metadata can be provided in the Information Object directly using the metadata label. In this case, the value can be any information. It is possible to tag the value using an attribute name. This is of special interest in order to allow both semantic searches by the user and to help NetInf managing objects.
- metadata (or a set of metadata) can be included from another pre-existing metalist. This functionality has been included in order to avoid users' redundancy, i.e. from providing several times the same list of metadata for a set of objects (for example, it makes possible the factorisation of the description of a set of pictures related to a single subject) and thereby increase the consistency of the description information.
- metadata (or a set of metadata) can be included from external metadata, i.e. metadata that are not in the metalist format. This can be the EXIF format [13] (the one used to store pictures' metadata in digital cameras) or any other as long as a source-to-source translator is available to automatically translate the metadata in the original format to the metalist format.

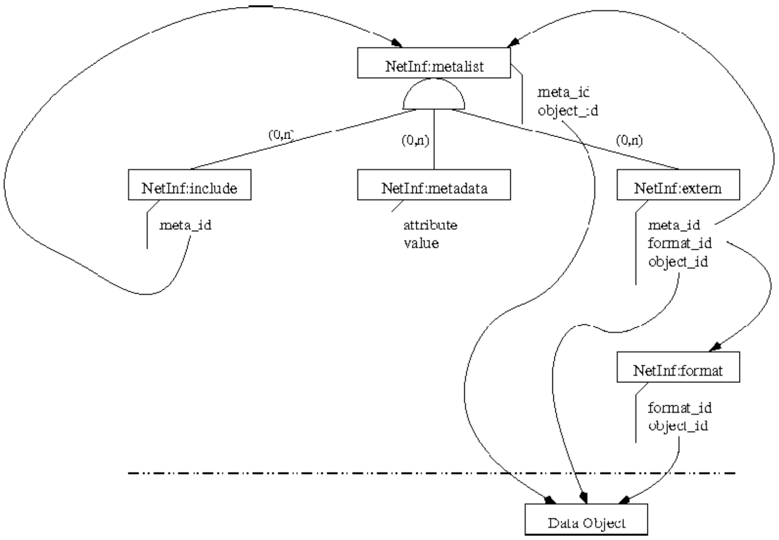

Fig. 2. The Information model

The prototype, developed in the scope of the 4WARD project [2], includes the implementation of the Metalist Model. XML was chosen as the support language even though other languages like JSON [3] would also have been acceptable. The description of the Metalist Model provided above in this section only aims at providing an idea of what functionalities the Information Model can provide. For further information [11] presents a complete description of the Metalist Model.

3.3 Global Architecture

The architecture of the Network of Information is based on three distinct spaces (see Fig. 3):

- The *Storage space* aims at storing all digital information related to the Future Internet. This can be objects directly, assuming these objects are digital ones, or the digital means to access real-world objects, for example a process to access an RFID tag.
- The *Index space* aims at storing the IOs provided by end-users so that the search for objects is performed more efficiently. In order to do so, an IO is preprocessed when presented to the index space and only the preprocessed version of the IO is stored in the index space.
- The *Communication space* stores nothing (at least from the NetInf point of view). It aims at providing a convenient mechanism to let the Network of Information, the Index space, the Storage space and the end-user exchange data. It can be based on any kind of protocol as long as some of the basic functionalities such as PUT and GET are available.

Note that if BOs are stored in the storage space, IOs are stored in both index and storage spaces using two different formats. Before being stored in the index space, IOs are processed in order to make them easier to search for. As such, IOs are not stored literally in the index space. Therefore, IOs are also stored in the storage space in order to allow end-users to refer to their description (for example to perform updates if they do not have any personal storage device).

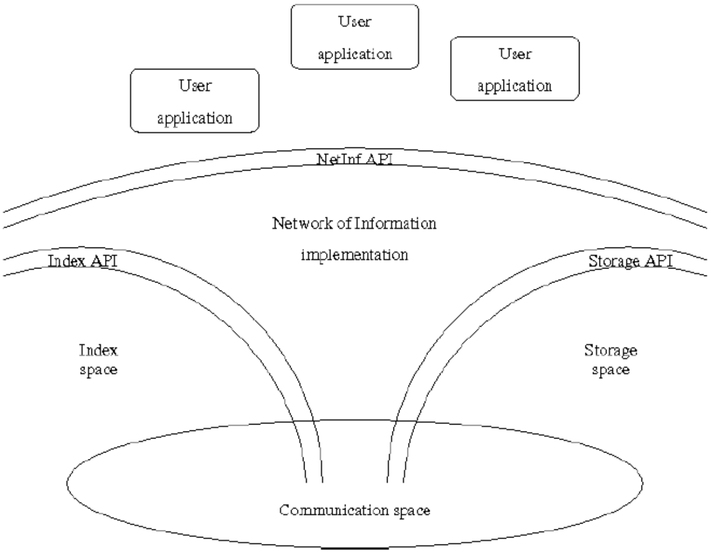

Fig. 3. NetInf global architecture

4. Secure Access to Objects

For a given object, access rights define both the type of rights and the users that are granted these rights. Thus, they are composing a set of metadata associated to the object. As NetInf is in charge of storing both the digital representation of objects and their description, these two kinds of objects are associated independent access rights. Typically, if the digital representations of objects are stored as a set of BOs, it is also the case with the raw representation of IOs which are, in fine, digital objects too. As a result, objects and their descriptions are accessible through two independent IOm. This is the reason why access rights, as metadata managed by NetInf, are stored in the IOm describing the object.

The main advantage of this organization is that the management of access rights for both objects and their descriptions are the same whatever the object type. This also means that users have to be very careful when manipulating access rights for IOs as this may definitely lead to a loss of access rights. However, we planed to provide a set of generic access rights for IOs that will be applied automatically when creating an object and updates of these access rights will be limited to very specific cases.

4.1 Principle

For each access type, a list of granted public keys is stored. If an access type is omitted, this means that the operation associated with this access type is open to the public. If an access type is present but no public key is provided, the meaning is that nobody is granted permission to perform the associated operation.

The access verification is performed by a *Security Controller* that is operating on the data of NetInf to check access rights for objects and their descriptions. For security reasons, it is important to locate the security controller in the storage system rather than in NetInf as this avoids a single party being able to manage all the security keys. The security controller is in charge of determining whether a user can access an object or not. It is not in charge of establishing the initial key agreement for the session.

This way, a user gains access to an object if the object is public or if the object is protected but the user was granted rights to access the object, i.e. the user holds a private key that matches a public key stored in the set of metadata devoted to security.

The security controller can be understood as a stand alone program that uses metadata devoted to security and is stored in NetInf to check users' access rights. Several instances of this program may be running in the system at the same time. In this case, all instances would run independently. There would be no race conditions between the different security controllers on accessing metadata devoted to security as security controllers are only reading these metadata and not writing them. The fact that a set of security controllers can be run concurrently allows the program to distribute them at any interesting locations depending upon the charge of the system and, as a result, ensures the scalability of the security mechanism.

For the sake of simplicity, once a secure channel has been set up for a user, the same security controller shall be used. This allows, for example, the caching of crucial information like the set of public keys that have been successfully challenged by the user.

4.2 Global View of NetInf Session

To allow a communication to take place to access data and/or metadata through NetInf, a classic two-step protocol has been envisaged: the first step (the key agreement step) consists of setting up a secured channel between the user and NetInf, and the second (the operation step) is a set of requests and responses exchanged between the two parties inside this secured channel. This two-step protocol is presented in Fig. 4.

The proposed schema is different from the Extensible Authentication Protocol (EAP) [14], since the user is not authenticated. However, the system checks if the user is holding a key that matches the requested access right. Moreover, EAP is associated with the physical layer and the schema proposed here is developed at the Application layer.

A key agreement step followed by an operation step is called a session. For a given session, more than one operation may be performed. The session ends when at least one of the two parties decides to retire from the session.

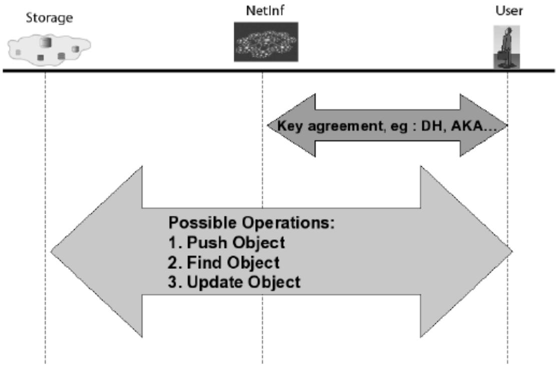

Fig. 4. Security protocol for NetInf

4.3 Key Agreement Establishment

Two hypotheses were used for the achievement of the key agreement between the user and NetInf. The first one is that no trusted third-party shall be used and the second is the possibility for the user to avoid revealing their identity.

As access to NetInf has to be performed securely, even the first access to NetInf should be secure. As a result, prior to any operation (typically at the creation of NetInf), NetInf must create a public/private key pair and release the public key.

When the user wants to connect to NetInf, he/she must first create a public/private key pair; then, he/she encrypts his/her own public key using the public key of NetInf. At reception, NetInf extracts the user public key from the encrypted message using its private key; then it uses the user's public key to encrypt messages to the user. This key agreement can benefit from already existing techniques like Diffie-Hellman and Elliptic Curve Diffie-Hellman ECDH [15], etc.

For example, [16] defines Elliptic Curve Diffie Hellman (ECDH) and Elliptic curve Menezes-Qu-Vanstone (ECMQV) as a 256-bit prime modulus for SECRET and a 384-bit prime modulus for TOP SECRET. Diffie-Hellman with a 1536-bit prime modulus as defined in [17] can also be used to produce a shared secret key of enough size for the key agreement in the insecure channel, the generator being 2.

As the user may change their public/private key pair for each session, the identity of the user cannot be revealed.

4.4 Example of Key Agreement Establishment

To establish a robust session between NetInf and the user, it is possible to use the protocol provided by McCullagh & Barreto [18]. The following presents their protocol as is. First, some preliminaries are provided, then, the core of the protocol is described. The goal here is to highlight the fact that not only do such protocols exist, but they are also quite easy to set up.

The conventional Tate pairing operates on a pair of points, P and Q, on an elliptic curve. The pair of points is denoted as $e\ (P;Q)$.

4.4.1. Preliminary

Let G_1 and G_2 be two groups; let P be the generator of G_1 with order prime q and Q be the generator of G_2 with the same order prime q. G_1 is a cyclic additive group and G_2 is a cyclic multiplicative group. Let a bilinear pairing be the following map $e : G_1 \times G_2 \quad G_2$ with the following properties [19]:

- Bilinearity: $e(ua, vb) = e(u, v)^{ab}$
  $e(u_1 + u_2, v) = e(u_1, v)e(u_2, v)$
  $e(u, v_1 + v_2) = e(u, v_1)e(u, v_2)$
  $p(au, v) = p(u, av) = p(u, v)^a$
  $\forall\ u_1, u_2, v_1, v_2 \in G_1 \text{ and } a, b \in Z_p$
- Non-degeneracy: $e(u, v) \neq 1$ for some $u, v \in G_1$
- Computability: there is an efficient algorithm to compute: $e(u, v)$ for any $u, v \in G_1$

A bilinear map satisfying these three properties is called an admissible map.

Tate pairing was introduced to cryptography [20] which came as an extension of the work proposed by Menezes.

4.4.2. McCullagh & Barreto's protocol

The protocol is triggered by a setup and an extract process which are well explained in [18]. In order to avoid getting too deep into details, only the key agreement process according to the view of NetInf and the User belonging to distinct domains is explained. By 'distinct domains' it is meant that both parties are referring to distinct authentication domains.

First of all, NetInf and the User must agree on the same elliptic curve and the same *(P;Q)* pair of points. Each of them generates his/her public/private key separately using any mechanism based on the elliptic curve, i.e. using the same elliptic curve and the same points *P* and *Q*. For example, NetInf can generate its public and private keys thusly:

$$NI_{priv} = (H(ID_{NI}) + S_{NI})^{-1}Q = (a + S_{NI})^{-1}Q$$
$$NI_{pub} = (H(ID_{NI}) + S_{NI})P = (a + S_{NI})P$$

Where IDNI can be any identity proposed by the NetInf, SNI is a secret key used by NetInf to generate any pair of keys at any time and for any session. H is a map to point hash function that map any hash value to a point on the elliptic curve.

In the same way, the user can generate his/her key pair as follows:

$$U_{priv} = (H(ID_U) + S_U)^{-1}Q = (a + S_U)^{-1}Q$$
$$U_{pub} = (H(ID_U) + S_U)P = (a + S_U)P$$

Where *IDU* is the identity of the user and *SU* is the secret key used by the mechanism adopted by the user to generate his/her key pair.

The procedure can then be described this way:

1. NetInf generates a random number *X*, computes $T_{NI\text{-}U} = XU_{pub}$, and sends it to the User.
2. The user does the same, i.e. she/he picks a random number *Y*, computes $T_{U\text{-}NI} = Y\ NI_{pub}$ and sends it to NetInf.
3. upon reception, NetInf computes:
$$\begin{aligned} &KNI = e(TU\text{-}NI , NIpriv)X \\ &KNI = e(Y.NIpub, (H(IDNI) + SNI) -1Q)X \\ &\quad = e(Y (H(IDNI) + SNI)P,H(IDNI) + SNI) -1Q)X \\ &\quad = e(P,Q)X.Y \end{aligned}$$
4. upon reception, the user computes the same value this way:
$$\begin{aligned} KU &= e(TNI\text{-}U, Upriv)Y \\ &= e(X.Upub, (H(IDU) + SU) -1Q)Y \\ &= e(X(H(IDU) + SU)P,H(IDU) + SU) -1Q)Y \\ &= e(P,Q)X.Y \end{aligned}$$

Note that the two parties could have the same shared value. In this case, this value is considered the shared key, according to McCullagh & Barreto. Also note that thanks to the identity of each party, the session key is generated. However, the identity of each party is not known to the other which is one of our assumptions.

4.5 Access to objects

After the session has been established, the user is authorized to access objects. Fig. 5 and Fig. 6 show the data flow between the different parties when accessing an object.

All data are stored in NetInf, and thus the access rights associated with objects are effectively stored in the storage space.

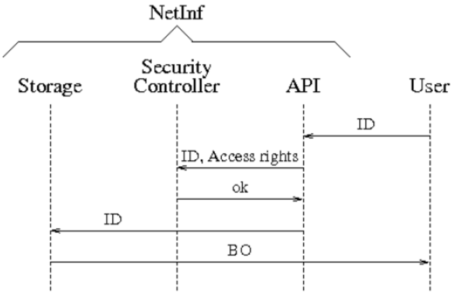

Fig. 5. Accessing a public object

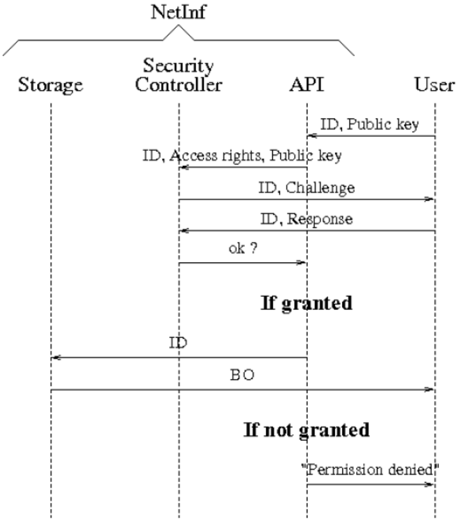

Fig. 6. Accessing a protected object

4.6 Optimizations

The use of public keys may add significant overhead to the communication. Typically, a single public key may require several kilo-bytes to be stored. If exchanged for every communication, this may overwhelm the network. Some optimizations may be performed to alleviate this potential problem. One consists of not requiring the public key itself, but a hash of the public key. It is then easy for the security controller to determine which public key is effectively held by the user. There is a very small probability that two public keys have the same hash, but this collision risk is acceptable, or the security controller may challenge all matching public keys. Another one consists of implementing a cache at the security controller level that stores a list of public keys that have already been presented by the user during the session.

When accessing a protected object, the user needs to provide a public key listed in the set of granted public keys together with the object ID. Then, the security controller challenges the user through NetInf which is granted access to the object if the response is correct, i.e. if the user effectively holds the private key associated with the presented public key.

All these operations are performed inside a secure channel between the user and NetInf, from the user point of view. From the NetInf point of view, there are two secure channels: the first one between the user and NetInf does not last longer than the session, and the second one between NetInf and the storage space can have an arbitrary duration.

It is the users' responsibility to determine which public key to present to the security controller to access an object. This can be easily performed by reading the list of public keys for a given access type in the metadata of the object or, if not available, to contact the owner of the object or any surrogate the contact information of which, if any, could also be stored in the metadata.

4.7 Access Rights

The way data and metadata are accessed and modified may be very different depending upon the intrinsic nature of objects. However, all can be expressed using the very same set of basic access rights. Thus, when designing the security architecture for NetInf, several possibilities to express access rights have been studied.

The first one is the Read-Write-Execute triplet of access rights together with the User-Group-Others triplet of user categories employed in the Unix file system. Despite the fact that this matches most of the basic cases and it has been widely used for decades, it remains very limited in expressing access rights. An extension introduced some years ago to enlarge the user categories and known as the Access Control List (ACL) does not provide any new rights to access files.

The second one is the set of rights managed by DataBase Management Systems (DBMS) through SQL Admin. It not only grants or denies any registered users a large variety of access types to the different components of a database (to create and drop tables, to select, update and delete records etc.), but it also grants some users the ability to grant others access thus allowing delegation of the whole management.

Even though the way access rights are managed in databases is far more complex than the Unix file system, some important aspects are still missing. For example, there is no way to include the notion of copyright [21]: how could a user provide a public object to the community and forbid the inclusion of this object inside another one? Some other types of access rights may be added in the future. This clearly depends upon the decisions that will be taken in the near future for the governance of the Future Internet. However, as access rights are metadata stored in NetInf, they have no impact on this security architecture and any other needed types of access rights should be added in the future.

Although, access rights are stored in the system using the Metalist model as described in [11]. The metalist model allows metadata to be provided as is for any objects, to be inherited from another description, or to be imported from another description facility.

4.8 Example of Access-Right Storage using the Metalists

Fig. 7 presents a part of an example of a Metalist (which ID is mid1) storing the security access rights of object oid1. Assume access-right types limit the operations on objects and their description to:

- access, the ability to read the content of an object.
- update, the ability to change the content of an object.
- annotate, the ability to provide a description of an object.
- remove, the ability to remove an object from NetInf.

In this example, no metadata with attribute access is specified. As a result, no control is performed when accessing the object. In other words, the object is publicly available. Note that as all other attributes are present in the metalist, the other access rights are protected.

For the access right to update, only one public key is provided. This means that a single user (probably the provider of the object) is allowed to perform any modifications on the object. For the access right to annotate, three public keys are provided. This means that only users holding the private keys associated with these three public keys have the right to annotate this object. Finally, the access right to remove is present in the metalist but no public key is provided. This means that nobody is allowed to perform this action on the object.

```
<NetInf:metalist meta_id="mid1" object_id="oid1" class_id="security">
<NetInf:metadata attribute="update">ecdh:2f6c69622f6c642d6c...696e</NetInf:metadata>
<NetInf:metadata attribute="annotate">ecdh:2f6c69622f6c642d6c...696e</NetInf:metadata>
<NetInf:metadata attribute="annotate">ecdh:75782e736f2e320004...0010</NetInf:metadata>
<NetInf:metadata attribute="annotate">ecdh:0100474e5502060861...6d24</NetInf:metadata>
<NetInf:metadata attribute="remove"></NetInf:metadata>
</NetInf:metalist>
```

Fig. 7. Storage of access rights using the Metalist model

5. Evaluation

According to [22], attacks may be divided into six categories depending upon the attacker’s goal, named as STRIDE: Spoofing Identity; Tampering; Repudiation; Information disclosure; Denial of service; Elevation of privilege. Depending upon the context, a single attack may result in different effects. However, the way attacks can be achieved is very limited. In the following, the different means used to perform attacks against our architecture are evaluated.

Eavesdropping (passive attack) This aims at listening to and looking for information exchanged between parties. In the scope of our architecture, this means listening to communications between the user and NetInf, NetInf and the storage space, or the user and the storage space. All these communications are embedded inside a secured session that can involve cryptographic operations. Therefore, it is not possible for any third party to access any content from exchanged messages between the different elements of our architecture.

Traffic analysis (passive attack) This aims at getting, from the flow of data, information like destination party, amount of transferred data, etc. In the scope of our architecture, this means analyzing communications between users, NetInf and the storage space. For the same reasons as above, the use of a secured channel does not allow getting information of any kind from messages transferred from one party to another. As a result, it is not possible for any third party to perform analyses to get statistics about the communications.

Replay (active attack) This attack aims at resending a message that has already been sent. In the scope of our architecture, this means that a malicious user would try to send a request to NetInf to get illegal access to data and/or metadata. There are two reasons that make this attack impossible. The first one is that no message can be retrieved from outside a secured session, and the second one is that the content of a message is completely different from one session to another. Thus, it is not possible for any third party to resend an already sent message.

Modification (active attack) This aims at intercepting data, changing it and injecting modified data in the message. In the scope of our architecture, this means analyzing communications between the parties and changing the contents of messages on the fly. For the same reasons as above, the use of a secure session to exchange messages does not allow any third party to inject new messages, or part of them, inside a pre-existing secured channel.

Virus Attack Three types of processes can be executed in the Storage System: user uploaded files, the process used to check compliance of a set of metadata to a class, the process used to import a set of metadata from an external format description. A malicious user could try to upload any file and run it as an executable file. If the file is not a regular executable file, it is harmless even though NetInf grants it execution rights. For example, if a malicious user uploads a JPEG image, any attempt to execute it will fail even if execution rights have been granted to the image. If the file is indeed executable, NetInf should use a sandbox to circumscribe any problems during the execution (like a virus attack) to the execution of the corresponding process and not to the entire system. The same is expected for both processes executed for checking metadata compliance to classes and importing metadata from external formats. This way, the system remains safe from virus attacks, even though they might be introduced by malicious users.

6. Challenging the Hypotheses

The work presented above is built on a set of hypotheses as described in Sec. 2. These hypotheses were chosen as they seem to be the most appropriate ones, i.e. the most stringent and realistic ones. However, the reality of the Future Internet may clearly be different from what can be foreseen as of today. As a result, this section aims at challenging the hypotheses to reveal the consequences of other choices.

No trusted third party According to the global organization of the Future Internet, any communications in the future will be performed through NetInf, i.e. NetInf will be the only intermediate agent or service to access the Internet. Therefore, any trusted third party should be accessed through NetInf which does not lead to a better security.

Another possibility would consist of having trusted third parties located outside of NetInf. In this case, the security would be limited to the ability of users to get the public key of the trusted third party securely the first time it is accessed, which is the same limitation as in the case where NetInf is accessed without any third party. As a result, using a trusted third party would not lead to better security for NetInf. Users may feel more secure in having it, but no security is added. However, this leads to a larger complexity for the implementation of NetInf as it requires including an external entity with which to work.

User identity not stored in the Network of Information It has been one of the major concerns to ensure the anonymity of users within the Future Internet even though allowing only granted users the ability to access and/or modify objects is definitively the main issue. However, user identity may be required for various reasons. For example, an author may be willing to make sure everybody knows she/he is the author of a given content; or an Internet provider may wish to leave access to authenticated users so as to protect itself against some provocative content published by some users. Whatever the reason, the solution remains the same: the identity of the author/owner should be stored inside the set of metadata attached to the object and signed to ensure nobody else but the owner effectively is provided the identity. In the former case, this shall be performed by the user as the Internet provider is not making the identity mandatory; in the latter case, this shall be performed by the Internet provider anytime an object is published or updated... Therefore, whenever there is an identity requested for the publication of data and metadata for the Future Internet, our approach remains the same.

Reliability of the Network of Information, the storage system and the routing The reliability on any of these three elements may be weakened if there is either a technical problem that reduces the security of the whole architecture or a malicious administrator (not an ordinary user but a user in charge of managing one of these three elements) that is willing to acquire un-granted access, reduce the capabilities of the system or even to destroy it.

Preventing malfunction from technical problems can usually be achieved by introducing redundancy. This would probably be efficient for the storage component, for example if the storage system uses a peer-to-peer organization. It would also be efficient for the routing component as it is today by re-sending messages that have not been received correctly. For NetInf, two cases have to be envisaged depending upon the number of NetInf systems that will be deployed in the future. If there are more than one NetInf, then redundancy should apply as equally as for storage and routing components but users will be in charge of managing this or a super-NetInf including the other ones shall be developed, leading to the other case. If there is only one NetInf, then it will be in charge of ensuring internal redundancy transparently to the user with multiple access points, duplication of metadata, etc.

Preventing malicious administrators from acting against the system is definitively the most difficult task, but there are some means to limit their malicious capabilities. NetInf can protect data and metadata from un-granted accesses by a malicious administrator located in the storage component or in the routing component using a combination of fingerprint, signature and encryption to store information, and providing only raw data to both storage and routing entities. This would not allow either the storage component or the routing component to associate data or a metadata to its fingerprint or signature. This solution is transparent to the user, event though it adds a significant overhead.

Protecting data from un-granted accesses by malicious administrators at any level (i.e. including NetInf) can be performed in the same way by the user. The user is then in charge of providing granted users the keys to access data. Note that a fingerprint and a signature can also be used to prevent NetInf from modifying metadata; however, metadata cannot be protected against reading by NetInf as NetInf requires this information to perform accurate search.

7. Conclusion and Future Works

This paper proposed a new solution to design a secure architecture for the Future Internet based on the NetInf Model. Based on a set of hypotheses the proposed solution is analyzed from a security point of view and then the hypotheses themselves are challenged. This in-depth analysis shows that no matter the governance issues that may arise in the future, our proposal remains consistent.

In future work, we will define the API so as to describe all the messages exchanged between entities. A simulator will also be implemented for the NetInf model and security parameters will be added to perform tests in realistic conditions. Emphasis and special focus on mobility and volatility issues are also planned since Future Internet's architectures will have to manage billions of mobile objects with discontinuing connectivity at the same time.

References

[1] C. Dannewitz, K. Pentikousis, R. Rembarz, E. Renault, O. Strandberg and J. Ubillos. Scenarios and Research Issues for a Network of Information. MobiMedia'08, July 7-9, 2008, Oulu, Finland.

[2] The FP7 4WARD Project. http://www.4ward-project.eu/

[3] D. Crockford. The application/json Media Type for JavaScript Object Notation (JSON). RFC 4627, Network Working Group, July, 2006.

[4] N. Niebert, S. Baucke, I. El-Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek and L.M. Correia. The way 4WARD to the creation of a future internet. PIMRC, 2008.

[5] R. Jain. Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation. Military Communications Conference MILCOM, October 23-25, 2006, Washington DC.

[6] D.D. Clark, J. Wroclawski, K.R. Sollins and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. SIGCOMM'02, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.

[7] J. Day. Patterns in Network Architecture: a Return to Fundamentals. Upper Saddle River, NJ, USA, 2008.

[8] C. Wang, A. Carzaniga, D. Evans and A.L. Wolf. Security issues and requirements for Internetscale publish-subscribe systems. Proceedings of the 35th Hawaii International Conference on System Sciences, 2002.

[9] Z. Miklos. Towards an access control mechanism for wide-area publish/subscribe systems. Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02), 2002.

[10] B. Ohlman et al. First NetInf Architecture Description. FP7 4WARD Deliverable 6.1. February, 2009.

[11] E. Renault and D. Zeghlache. The Metalist Model: a Simple and Extensible Information Model for the Future Internet. EUNICE 2009 - The Internet of the Future, September, 2009, Barcelona, Spain.

[12] BitTorrent. http://www.bittorrent.com/

[13] Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.2. Standard of Japan Electronics and Information Technology Industries Association, April, 2002.

[14] L. Blunk and J. Vollbrecht. PPP Extensible Authentication Protocol (EAP). RFC 2284, March, 1998.

[15] X9.62-1998 (draft). Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). 1998.

[16] E. Barker, D. Johnson and M. Smid. Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. NIST Special Publication 800-56A, March, 2007.

[17] T. Kivinen and M. Kojo. More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key exchange (IKE). RFC 3526, May, 2003.

[18] N. McCullagh and P.S.L.M. Barreto. A new two-party identity-based authenticated key agreement. In Cryptographers Track at RSA Conference, 2005.

[19] A.G. Myasnikov. Theory of models of bilinear mappings. Omsk City. Translated from sibirskii Matematicheskii Zhurnal, 31(3):94-108, 1990.

[20] G. Frey and H.-G. Ruck. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. Math. Comp, 62(206):865-874, 1994.

[21] S. Rao. Copyright: its implications for electronic information. Online information review, 27(4):264-275, 2003.

[22] X800 CCITT Recommandation: Data Communication Networks: Open Interconnection(OSI); Security, Structure and Architecture for Open SystemsInterconnection for CCITT. 1992.

Éric Renault

Éric has been an associate professor at the Institut Télécom – Télécom SudParis, Évry, France, since late 2001. His research interests include cluster and grid computing, high-performance messaging together with low-cost security and the development of the Future Internet. He has been involved in several national and European projects. Éric received an MSc in Computer Engineering and an MSc in Computer Science in 1995 and a PhD in Parallel Computing in 2000 from the University of Versailles Saint-Quentin-en-Yvelines, France, where he also served as an assistant professor. In 2001, he was a research associate at Dartmouth College, NH. Éric is the author of more than 50 papers published in peer-reviewed journals and conferences.

Ahmad Ahmad

Ahmad received in 2000 a Bachelor degree of Electronic and Communication Systems Engineering from Tishreen University - Lattakia, Syria. In 2001, he received a Postgraduate Diploma in Communication Systems from the same university. In 2005, he received a Postgraduate Diploma in communication systems and networks from the University of Montpellier II, France. He already had a PhD degree in Security of Wireless Networks from Institut Télécom – Télécom SudParis, France, 2010. During his PhD studies, he worked on the MAGNET, MAGNET Beyond and 4WARD projects. After his PhD, he joined Institut Télécom – Télécom ParisTech, France, where he is currently a Post-doc within the INFRES department.

Mohamed Abid

Mohamed is currently a PhD Student in his third year (started in 2006) in Institut Télécom – Télécom SudParis, France. He holds a Master’s Diploma in Computer science in 2005 from ENSI Ecole Nationale des Sciences de l’Informatique, Tunisia. His research interests include Control access in networks (network security), Biometric and cryptography, Future Internet, and Handover in wireless networks.