Data-based access control in Named Data Networking

Balkis Hamdane[1,2], Mounira Msahli[1], Ahmed Serhrouchni[1],Sihem Guemara El Fatmi[2],
[1]Télécom ParisTech, Paris, France
[2]Higher School of Communications of Tunis, Ariana, Tunisia
{balkis.hamdane, mounira.msahli, ahmed.serhrouchni}@telecom-paristech.fr
sihem.guemara@supcom.rnu.tn

***Abstract*— Named Data Networking (NDN) presents one of the first and most emergent Information Centric Networking (ICN) project. It offers an excellent substrate to solve today's Internet problems. To ensure security challenge, it adopts a data-centric model. The access control represents a fundamental security aspect. It prevents the data publication under any sensitive namespace and the access to any confidential content. In this paper, we use the generic and conceptual access control scheme called UCONABC to propose an optimum and secured data centric access control model. In our proposal, data is protected by encryption and lock password. Its access is managed by a centralized access list (ACL).**

***Keywords*—Access control, Content Centric Networking, Named Data Networking, UconABC.**

I. INTRODUCTION

Information Centric Networking (ICN)[1] constitutes a promising direction for the Internet of the Future research. Content Centric Networking (CCN)[2] presents one of the first and most emergent ICN projects. It was proposed by Van Jacobson in 2006 in the Palo Alto Research Center (PARC). In September 2010, it was chosen as one of the National Science Foundation's Future Internet Architecture (FIA) projects. It is called in this new context Named Data Networking (NDN)[3].

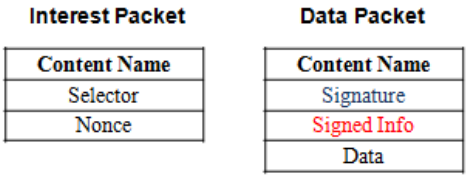

Figure 1. NDN packet Types

In this project, named data represents the central element; all network operations are determined by content name rather than IP address. They are based on two types of packets: Interest and Data (see Fig. 1). Interest packet is sent to the network by consumer to request content. It carries a hierarchical and human-readable name identifying the desired content. Data packet represents the response to the Interest. It is characterized by the same name in the Interest packet (the name in the Interest packet may also represent the prefix of the name in the Data packet). It contains also a signature calculated on the entire packet (name and data) using the data producer private key. This signature securely binds the name to the data. To verify it, the requester recovers the producer public key based on information provided in the field signed info (data producer public key digest…).

Through in-network caching, Data packets can be sent by any node receiving the Interest and having the requested data. Same named content can reside in various network locations and a request can be satisfied by the closest copy of the content. However with this content dispersion, security can no longer be tied to a particular host. Therefore, a data-oriented security model must be adopted. NDN offers an excellent substrate for achieving the requirements of such security model. To deal with integrity protection and data origin authenticity, it makes mandatory the digital signature of each data and its corresponding name, securely binding them. To achieve other security goal, NDN can be extended using new or traditional mechanisms.

The access control represents a fundamental security aspect since publishers need to control access to their contents. In this paper we use the UCONABC[4] model to define access rights management in NDN networks. As mentioned in [4], the UCONABC model defines six generic components: subject(S) which is an entity associated with attributes or ATT(S) such as (user id or user name, etc...). Object (O) is a set of entities on which subjects have rights characterized by its attributes ATT(O). Rights are privileges that a subject can hold over an object such as (read and write). Authorizations are functional predicates that must be evaluated to determine the possibility of use of an object by a subject. Obligations are checked functional predicates that have to meet mandatory requirements. Conditions are decision factors.

In this paper, we propose a data-based access control model. In this model, protected content are encrypted and protected using a password. In addition, access rights are organized around the concept of namespaces [5] and they are described in an access control list (ACL).

The map of this paper is as follow: in the second section we describe our scheme, we detail our access rights management model and we represent readers operations and writer

COLLABORATECOM 2013, October 20-23, Austin, United States

DOI 10.4108/icst.collaboratecom.2013.254180

operations. In the third section we give the related work and finally we conclude in the fourth part.

II. OUR SCHEME

Access control is an important aspect of security in information centric networking approach. If no access control is applied, there will be no distinction between legitimate and malicious entities. Publishers will be able to publish data under any namespace and requester can access to any content.

The importance of this security aspect led us to propose an access control model. Our proposal is based on data locking allowing protected content to freely reside anywhere in the network. Only legitimate readers can unlock protected data and only legitimate writers can recover the parameters necessary to the publication of a locked data.

The proposed solution is organized around the concept of namespaces. Indeed, data are characterized by hierarchical and tree-like structure, so in order to manage large collections of data, a sub-tree of names can inherit the same policy access control as that of the root node. On the other hand, to provide more flexibility, an arbitrary assignment of access rights to a child node is possible.

To control the access to a namespace, an access control list (ACL), an asymmetric root node key pair ($RNK_{encryption}$, $RNK_{decryption}$) and a password are created. The ACL specifies access rights (read or write) of the authorized entities for this namespace. It is composed by a list of legitimate groups, their rights and links that point to their public keys.

The $RNK_{encryption}$ key encrypts the symmetric data key K_{data} used to encrypt the protected data associated to namespace and the $RNK_{decryption}$ key is used for its decryption.

A trusted server $S_{authorization}$ is directly attached to the root node of a namespace. It communicates and exchanges data about the ACL securely with this node. $S_{authorization}$ maintains a list composed of legitimate and revoked groups, their rights, dates of the attribution or revocation of these rights and passwords associated with these dates. In addition, it is responsible of the mapping between the group identifiers in the ACL and the public keys digests of the members of these groups. It manages also the reading and writing authorizations based on information retrieved from the ACL (see Fig. 2).

At least one ACL and one authorization server are associated with a protected namespace. They permit to control the access to only resources published in this namespace.

A. Access rights management

To assign the different access rights of a subtree, the entity E holding the private key associated with the root node creates an ACL containing the various rights of entities. In addition, E encrypts the $RNK_{decryption}$ key using the public key of every group with only a read access and encrypts the key pair ($RNK_{encryption}$, $RNK_{decryption}$) using the public keys of every group with a read and write privileges. It publishes these encrypted keys in Data packets (carrying a signature calculated using its private key). On the other hand, the trusted server $S_{authorization}$ provides passwords encrypted with the public keys of authorized entities which are required to perform read and write operations.

To handle inheritance of access rights on nodes, each child node inherits the same key pair ($RNK_{decryption}$, $RNK_{encryption}$) and the password from the root node.

To affect new rights to a child node N, a new ACL indicating new access rights, a new key pair ($RNK_{decryption}$, $RNK_{encryption}$) and a new password are assigned to it. Also, a new $S_{authorization}$ server, managing the password attribution and the reading/writing authorizations, is attached to this child node. Changes of rights apply to all the children of N.

To revoke a reading and/or the writing privilege of a group for a particular namespace, the password must is revoked. A new random one is created and encrypted with the public keys of non revoked entities. The ACL is also be updated and the Sauthorization server saves the revocation date of the access right and the old password. The same key pair ($RNK_{encryption}$, $RNK_{decryption}$) and the new password will then be used by the unrevoked entities to read or write a new content. For an old content, authorized readers still use the same key $RNK_{decryption}$ and recover the old password used during the content creation from the authorization server. The use of passwords is then useful to avoid the revocation of all encrypted keys and the generation of new ones encrypted with the public keys of authorized groups.

To add a read (and write) right to a group, the RNKdecryption key (and $RNK_{encryption}$ key) is encrypted using its public key. Also, the ACL is updated and the $S_{authorization}$ server saves the attribution date of the access right.

Using UCONABC, the $S_{authorization}$ access list is given by the model bellow:

S: is a user

R: is a right = {read, write}

O: is data

G is a set of groups of subject S

$groupId: S \rightarrow 2^G$ *many to many mapping*

$ACL: O \rightarrow 2^{G.R}$, *g is authorized to do r to o*

$ATT(S): \{groupId\}$

$ATT(O) = \{ACL\}$

$allowed\ (S, O, R) => \{(g, r) | g\ in\ groupId(s) \cap ACL(o)\}$

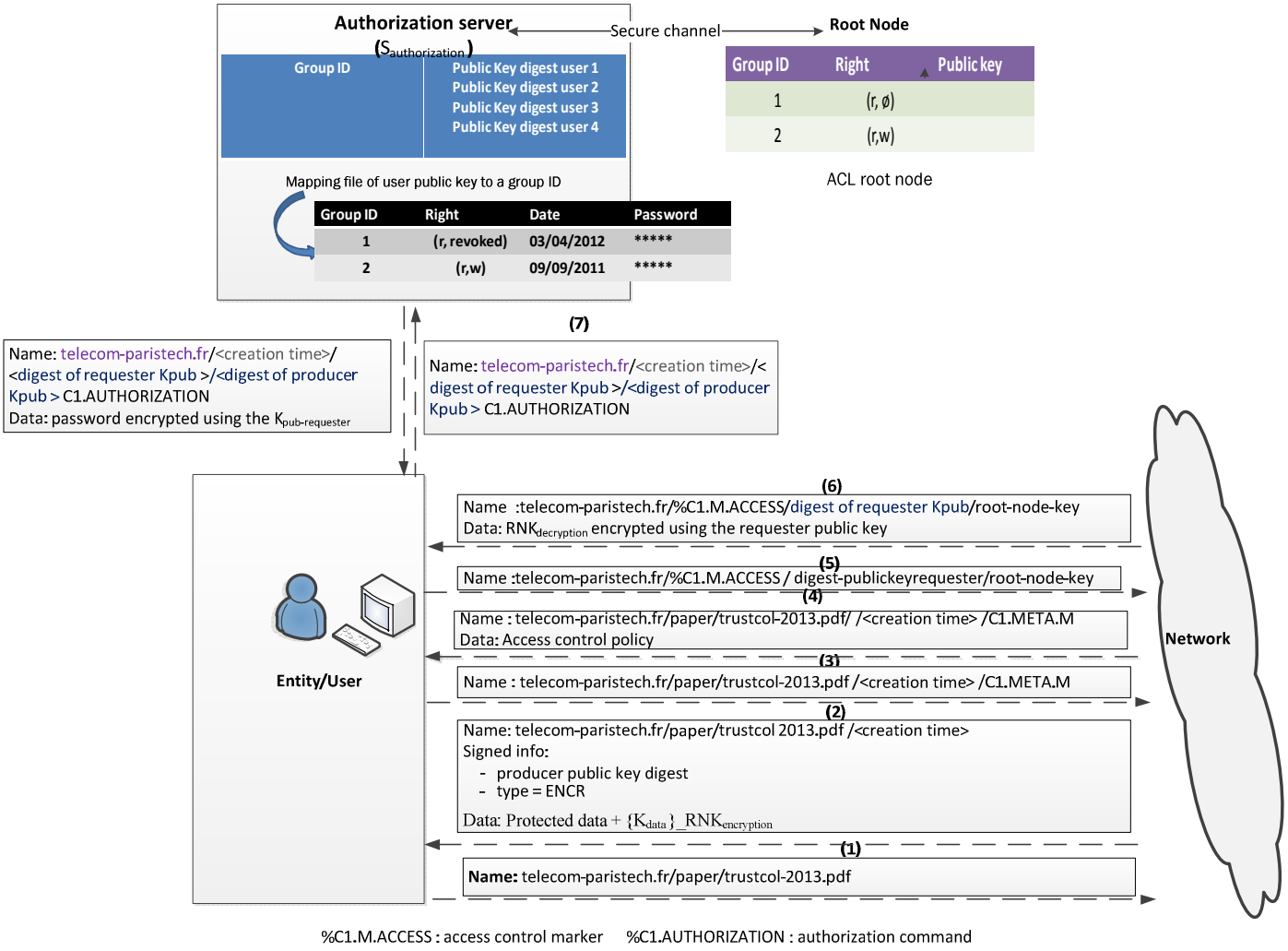

Figure 2. Reader operations

In order to authorize access to a namespace, three conditions must be checked: use of a valid password to unlock data, unrevoked root node and unrevoked entity. Using UCONABC, access control conditions are modeled by:

S: is an entity

O: is data

R: is the right

$ATT(R) = \{read, write\}$

$Condition = \{(valid\ password).(non\ revoked\ root).(non\ revoked\ entity)\}$

getCON =

$$\begin{cases} (password, valid)\ if\ password = OK \\ (root\ node, non\ revoked)\ root\ revoked\ date = valid \\ (entity\ node, non\ revoked)\ root\ revoked\ date = valid \end{cases}$$

$\text{allowed(S, O, R)} => Concheked(getCON(S, O, R))$

B. *Reader operations*

When a user tries to read a protected content, he requests it by sending an Interest packet (packet (1) in Fig.2). He receives the corresponding Data packet (packet (2) in Fig.2) and discovers that it is encrypted and protected by a password (marked as type ENCR). The content name contains the creation time of this packet and the Data field is composed of desired protected content and a symmetric data key K_{data} encrypted with the $RNK_{encryption}$ key. The consumer asks then for metadata of this namespace ((packet (3) in Fig.2). He retrieves the access control marker indicating the root node of the protected subtree and referring to the used naming conventions (packet (4) in Fig.2). Based on these conventions, he derives the name of the root node key encrypted with the public key of the group to which he belongs. He requests it (packet (5) in Fig.2). If he has only a read access, he receives $RNK_{decryption}$ encrypted with his public key (packet (6) in Fig.2).

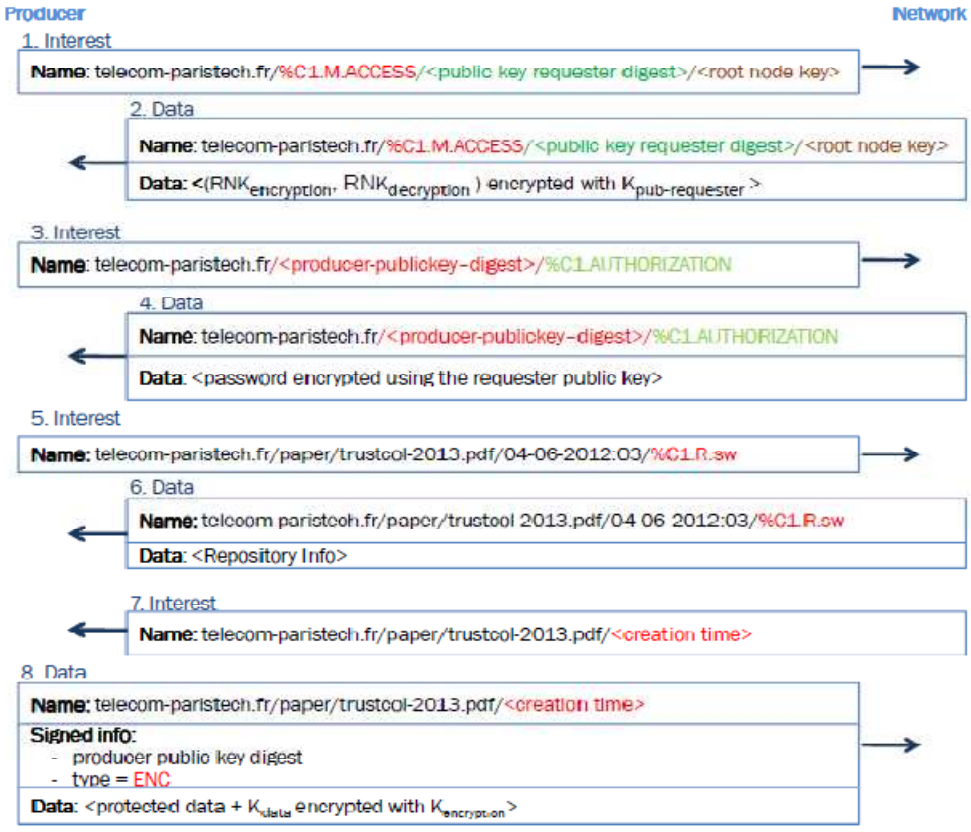

Figure 3. Writer operations

If he has also the write privilege, he retrieves the key pair ($RNK_{encryption}$, $RNK_{decryption}$) encrypted. He uses his group private key to derive root node key and employs it after to decrypt the data key K_{data}. After that, the requester deducts the name of the root node, the data creation time from the content name and the data producer public key digest from the signed info field. He sends an Interest packet with a content name composed by the root node name concatenated to the public key digests of the producer and the requester, the creation time of the protected data and an authorization command (packet (7) in Fig.2). When this packet arrives to the trusted server $S_{authorization}$, this entity looks for a mapping between the public keys digests of the requester and the producer and their corresponding group identifiers. Once these group identifiers are found, the list containing statutes of various entities is examined for the first time to check the current read right of the requester. If this right is verified, this list is reviewed a second time to check the write permission of the producer at the creation time of the protected content and to retrieve the used password. After this checking, the password needed to read the content is sent encrypted using the public key of the requester group (packet (8) in Fig.2). The user has now all the parameters necessary to read the protected data. He uses the password to obtain the data encrypted with the data key K_{data}. This key is then decrypted by the $RNK_{decryption}$ key and used to decrypt the desired data.

If the requester or the producer does not have the required rights, or if one of the mapping between the public keys digests of the requester and the producer and their corresponding group identifiers is not found, a data packet signaling an access denied error is sent .

Fig. 2 illustrates the operations performed by user having a non revoked read right for the protected namespace “telecom-paristech.fr”.

C. Writer operations

To write a piece of content with a restricted access, a user requests the key of the root node encrypted with his group public key. He can retrieve the $RNK_{encryption}$ key only if he has the write privilege for the namespace. He uses his group private key to decrypt it. After that, he sends an Interest packet to request the password. This packet contains a content name composed of the root node name, his public key digest and an authorization command (“%C1.AUTHORIZATION”). It is routed towards the corresponding server $S_{authorization}$ (it contains the authorization command). When this server receives this request, it checks its list and verifies the write privilege of the producer. If this producer has the required right, the password needed to protect the content is sent encrypted with his public key. By receiving this password, the user first produces a symmetric key K_{data} and utilizes it to encrypt the content. He then encrypts the K_{data} key with the $RNK_{encryption}$ key and he

protects the encrypted data with the received password.

After that, he requests to save the protected data by sending an Interest packet with the start write command "%C1.R.sw" in the content name. He receives information on the backup directory and finally he publishes the Data packet marked as type ENCR. The content name of this packet contains its creation time, and the data field is composed of the protected data and the encrypted K_{data}. All write operations are shown in Fig. 3 where the root node of the controlled subtree is named "telecom-paristech.fr".

III. RELATED WORK

NDN offers an excellent flexible platform to ensure security challenges. Several proposals were made to improve this aspect [6, 7]. However, the only access control model is proposed in [5]. This solution is developed in the current NDN prototype CCNx [8]. It is based on data encryption and is organized around the concept of namespaces.

To restrict the access to a namespace, an ACL and a symmetric node key (NK) are created and associated with the root node. The NK key is employed to derive the symmetric data keys (DK) used to encrypt data and it is encrypted with the public key of every entity in the associated ACL.

To handle inheritance, the access control solution proposed in CCNx adopts the same modified hierarchical access control model. However, the node key of children node is derived from the NK key of their parents using a key derivation function.

For rights revocation, the update of the ACL is sufficient to apply a write or management access changes. The read access revocation needs also the generation of a new node key NK' for the root. NK' is encrypted with the public key of authorized entities and the associated children node keys can be directly derived from NK'.

To add an access right to an entity, the corresponding ACL is updated and the associated node key is encrypted under the key of the added entity.

To read a protected content, a consumer receives a Data packet and discovers that it is encrypted (type= ENCR). He then requests the data key DK encrypted with the node key NK. After that, he retrieves the node key and uses it to decrypt the DK. He finally uses the later key to decrypt the desired content.

Read operation seems simple. However, if the user wants to ensure the write privilege of the content producer, he needs to perform many operations. He should first ensure the management right of the ACL's signatory by requesting the ACL associated with the node (or its parent node) as well as all its previous versions and checks them. After, he checks the latest version of the ACL for the validation of the content producer right. If the producer of the content or of the ACL belongs to a group, additional work must be done to find and validate their group membership.

To write a piece of content with a restricted access, a producer requests NK key encrypted with his public key and decrypts it. After, he generates a random data key DK and uses it to encrypt the content. He uses NK key to encrypt the generated key. Finally, he publishes the DK encrypted with the key node NK and the protected content. The described solution is characterized by certain limits. Indeed, a user with only a read right for a node can recover the associated NK key. He can use it to encrypt a data key DK and correctly encrypt a content with a DK key. The discovery of this attack requires firstly performing all necessary operations to recover and decrypt the content (the retrieval and the decryption of the DK and NK keys and the content). The user must also retrieve all the versions of the ACL associated with the node (or its parent node). He should examine them to find the rules that relate to the producer. Finally, he must verify the ACL associated with the content to verify the producer write privilege.

Another problem resides in the vulnerability to the write rollback attack; an entity with a revoked write access can write new content and can claim to have been written at a previous time. Finally, in this solution the ACL can be clearly exchanged revealing personal information on the access rights of the various entities.

Our solution keeps some characteristics from the solution described in [5]. Indeed, both solutions are organized around the concept of namespaces since all network operations are based on content names. They use a symmetrical encryption considering its speed of execution. They also adopt a modified hierarchical access control model which is at the same time flexible and useful to manage large collections of data.

However, our proposal eliminates the disadvantages of the solution described above. It prevents an entity with only a read access to write since this operation requires the $RNK_{encryption}$ key (known only by the authorized entity). To bypass the rollback attack, the data creation time is added to the content name of the data packet containing the protected content. The authorization server compares before any authorization the date of creation and of the right revocation. Finally, the ACL is kept secret which prevents the disclosure of personal information about entities.

IV. CONCLUSION

NDN project represents a major Information Centric Networking candidate for Future Internet architecture. It focuses on solving today's Internet limits related to mobility, routing, and security. To ensure access control, it adopts a data-based model since same named content can reside in various network locations.

In this paper, we provide a new optimum access control

model in NDN. We use a conceptual paradigm called the UCONABC to define access rights management over NDN. In our model, content is encrypted and locked using a password. In addition, access rights are based on the concept of namespaces and described in an access control list (ACL).

Our access control solution fits in perfectly with the NDN project and it does not require any change in NDN structures which makes its validation implicit. Our current and future work focus on the validation of this proposal through its implementation in CCNx.

REFERENCES

[1] BengtAhlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, BorjeOhlman . (2012). A survey of information-centric networking. *IEEE Communications Magazine 50(7): 26-36*.

[2] V. Jacobson, D.K. Smetters, J.D. Thornton, M. Plass, N. Briggs and R. Braynard, " *Networking named content*", in Proceedings of the 5th ACM International Conference on Emerging Networking Experiments and Technologies, (CoNEXT 2009). Rome, Italy , 2009 December 1-4;. pp. 117‑124.

[3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos and al (2010,October). Named data networking (ndn) project. Technical report NDN-0001, Xerox Palo Alto Research Center PARC. [Online].Available:http://www.named-data.net/techreports.html

[4] Jaehong Park, Ravi Sandhu, The UCONABC usage control model, ACM Transactions on Information and System Security (TISSEC), February 2004.

[5] J. T. Philippe Golle and D. Smetters. CCNx access control specifications. Technical report, Xerox Palo Alto Research Center-PARC, 2010.

[6] B. Hamdane, A. Serhrouchni, A. Fadlallah and S. Guemara, "*Named-data security scheme for named data networking"*, in Proceedings of the third International Conference on the Network of the Future (NoF 2012). Tunis, Tunisia, 2012.

[7] Xinwen Zhang, Katharine Chang, HuijunXiong, Yonggang Wen, Guangyu Shi and Guoqiang Wang, *"Towards name-based trust and security for content-centric network"*, in Proceedings of the 19th IEEE International Conference on Network Protocols (ICNP 2011): Vancouver, BC Canada, October 2011.

[8] Project CCNx home page. Available : http://www.ccnx.org/.