NAC: Name-Based Access Control in Named Data Networking

Zhiyi Zhang, Yingdi Yu, Alexander Afanasyev, Lixia Zhang University of California, Los Angeles, CA, USA
Email: {zhiyi,yingdi,aa,lixia}@cs.ucla.edu

***Abstract*—As a proposed Internet architecture, Named Data Networking must provide effective security support: data authenticity, confidentiality, and availability. This short paper focuses on supporting data confidentiality via encryption. The main challenge is to provide an easy-to-use key management mechanism that ensures only authorized parties are given the access to protected data. We describe the design of name-based access control (NAC) which provides automated key management by developing systematic naming conventions for both data and cryptographic keys. We also discuss an enhanced version of NAC that leverages attribute-based encryption mechanisms (NAC-ABE) to improve the flexibility of data access control and reduce communication, storage, and processing overheads.**

I. Introduction

Effective security support for data authenticity, confidentiality, and availability is an important requirement for Named Data Networking (NDN) [1]. In this short paper, we present the design of name-based access control (NAC) for NDN, which provides an automated key management mechanism for content confidentiality through encryption. Note that complete communication confidentiality in an NDN network requires both *content* confidentiality and *name* confidentiality. We address the former here and the latter in future work.

The main idea of NAC is simple. Producers of confidential data encrypt data when data is produced, and the access control system ensures that only authorized consumers can obtain the keys needed to decrypt it. This design eliminates the reliance on intermediate parties (e.g., data storage, firewalls, or routers) to enforce access control. However, a security solution will get used only if it is easy to use. One way to achieve this is through automation of the cryptographic key management. We explore the use of NDN naming conventions to develop systematic ways to name encrypted data and the related keys, so that legitimate consumers can securely retrieve decryption keys without needing any additional information other than the names of their desired data.

Below we present the basic NAC design (Section II), followed by its extension, dubbed NAC-ABE, to support attribute-based encryption [2]. NAC controls data access granularity through the conventions of hierarchically structured names. NAC-ABE extends this concept to provide further control based on the semantics of named attributes, which may also be hierarchically structured.

The basic version of NAC has been implemented as a stand-alone C++11 library [3] and integrated into NDN-CCL library suite [4]. An initial prototype of NAC-ABE extension is implemented as a separate C++11 library [5].

II. Name-Based Access Control (NAC)

One can use NDN's namespace to convey rich contextual information for data access control. For instance, one may name the write access control key for publishing data under a given prefix "`/example/_WRITE`", and the read access control key under the same prefix "`/example/_READ`". One can also use the hierarchically structured namespace to support access control with fine granularity, e.g., "`/example/sub_space/_READ`", "`/example/sub_space/_WRITE`". We call such systematic naming rules *naming conventions*. NAC consists of a system model and a set of specific naming conventions.

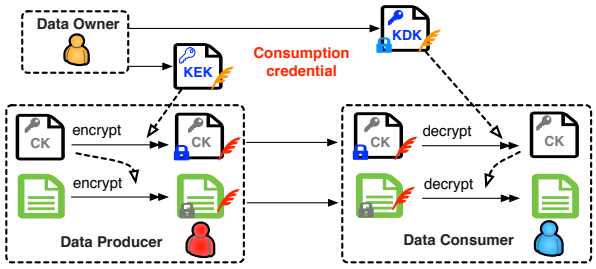

Fig. 1. Major system components in NAC

NAC System Model. Figure 1 shows the overall system structure for NAC, which consists of three types of entities: data consumer, data producer, and data owner.[1] The data owner provides two types of credentials: *production credential* and the *consumption credential*. A production credential, which is not covered here, authorizes a legit producer to produce and sign data under a given namespace N. A consumption credential is a pair of public/private keys generated by data owner, called KEK (key encryption key)/KDK (key decryption key), respectively, that are used to control the access to the content under namespace N in the following way. First, a producer generates a symmetric key (*content key*, CK) and uses it to encrypt its content. Then the producer uses the data owner's KEK to encrypt CK (which can only be decrypted by the data owner's KDK). The data owner securely passes the KDK to *each* authorized consumer U by using U's public key to encrypt the KDK. Both the encrypted content key CK and encrypted KDK are published, so that U can retrieve and decrypt CK.

Naming Convention. NAC names all data packets carrying encrypted content by the naming convention as shown in Figure 2, where "`ENCRYPTED-BY`" is a special tag component. This naming convention uses a data packet's name to carry, as a suffix, the name of the key used to encrypt its content. Thus when one retrieves a data packet by its standard name, if the

[1]The separation of data owner from producer is needed in cases where producers are resource constrained devices, such as small sensors, which cannot directly execute access control functions; they can be the same entity for powerful data producers.

requested content is encrypted, the returned data packet has a longer name which contains the name of the encryption key.

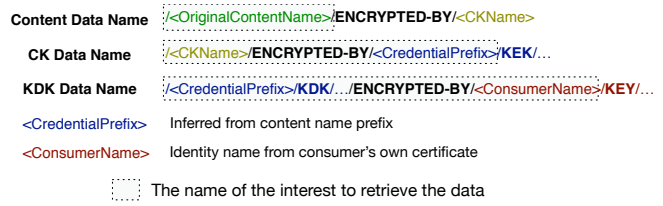

Fig. 2. Naming conventions of NAC

Naming conventions can also be used to set the access granularity needed by a given application. To illustrate, we show an example implemented in our codebase. In addition to the hierarchical namespace which identifies the data source, two timestamp components are added to the data and key names, to control data access by specific time periods. For instance, "`/example/_READ/data/ENCRYPTED-BY/example/_READ/CK/20170713/20170714`" cannot be decrypted by the CK whose suffix is "`/20170712/20170713`". Further details are omitted here due to the space limit.

III. NAC-ABE

To provide additional flexibility for access control policies, we extend NAC with ciphertext-policy attribute-based encryption (CPABE) [6], [7] to create NAC-ABE. In this extension, a data owner still retains the control over the data production path, but the consumption control is delegated to an attribute authority (Figure 3).

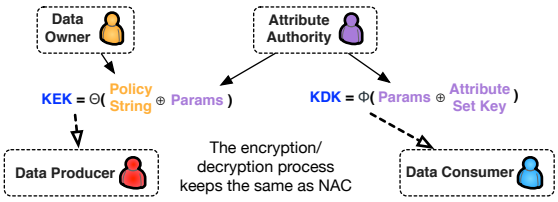

Fig. 3. Major system components in NAC-ABE extension

More specifically, a data producer in NAC-ABE encrypts a data packet with a owner-defined policy key, which are derived from the public parameters of the attribute authority (usually referred to as *params*) and a plain-text attribute string or a combination of attribute strings and conditional statements, such as "student", "register-year $>$ 2014", or "UCLA *and* student." To decrypt the policy-encrypted data, a consumer U must own sufficient attributes ("UCLA", "student") to satisfy the policy ("UCLA and stduent") in the form of a cryptographic key (representing attribute set: "UCLA", "student") issued by an attribute authority.[2] The naming convention of NAC-ABE, shown in Figure 4, allows U to directly understand whether the attributes that U owns, or can obtain, are sufficient to access the desired data.

[2]For simplicity, in the following description we assume that the attribute authority can reliably verify which attribute(s) each consumer U possesses, and upon request, issue keys to U for the set of attributes U has; further details are omitted here.

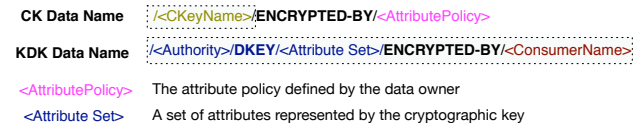

Fig. 4. Naming conventions of NAC-ABE extension

Attribute Authority as One Level of Indirection. In the basic NAC, data owners directly manage the data access, verifying each consumer's identity and encrypting KDKs for all authorized ones. This design raises scaling concerns to both data owners (when the number of consumers gets large, a data owner has to encrypt the KDK for everyone of them), and data consumers (when one collects data from a large number of sources, one has to handle a large number of decryption keys). NAC-ABE allows data owners to simply define attributes needed to access their data, and data consumers to have sets of attributes to obtain the decryption keys. The attribute authority's role is to properly vet the consumers and provide decryption keys to those with attested attributes.

IV. Conclusion and Future Work

The design of NAC provides a general approach to provide data confidentiality and access control in Named Data Networking. Some engineering optimizations for security, performance and robustness are omitted in this short description. Additionally, access rights revocation and name confidentiality remain as the future work.

ACKNOWLEDGMENT

This work is partially supported by the National Science Foundation under awards CNS-1629922 and CNS-1719403, as well as by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named Data Networking," *ACM SIGCOMM Computer Communication Review*, 2014.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. of Conference on the Theory and Applications of Cryptographic Techniques*, 2005, pp. 457–473.

[3] Y. Yu, A. Afanasyev, and L. Zhang, "Name-based access control," Technical Report NDN-0034, Revision 2, 2016.

[4] J. Thompson *et al.*, "NDN-CCL API," https://github.com/named-data/NDN-CCL-API, 2017.

[5] Z. Zhang and Y. Tu, "NAC-ABE codebase," https://github.com/Zhiyi-Zhang/NAC-ABE, 2017.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. of IEEE Symposium on Security and Privacy*, 2007.

[7] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in icn: Attribute-based encryption and routing," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, 2013.