NAC: Name-Based Access Control in Named Data Networking

Zhiyi Zhang
UCLA
zhiyi@cs.ucla.edu

Yingdi Yu
UCLA
yingdi@cs.ucla.edu

Alexander Afanasyev
Florida International University
aa@cs.fiu.edu

Jeff Burke
UCLA
jburke@remap.ucla.edu

Lixia Zhang
UCLA
lixia@cs.ucla.edu

ABSTRACT

As a proposed Internet architecture, Named Data Networking must provide effective security support: data authenticity, confidentiality, and availability. This poster 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.

CCS CONCEPTS

• **Security and privacy** → **Access control**; • **Networks** → Naming and addressing;

KEYWORDS

NDN, access control, ABE

ACM Reference Format:
Zhiyi Zhang, Yingdi Yu, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. 2017. NAC: Name-Based Access Control in Named Data Networking. In *Proceedings of ICN '17, Berlin, Germany, September 26–28, 2017,* 2 pages. https://doi.org/10.1145/3125719.3132102

1 INTRODUCTION

Effective security support for data authenticity, confidentiality, and availability is an important requirement for all proposed Internet architectures, including Named Data Networking (NDN) [6]. In this poster, 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 generated and the access control system ensures that only authorized consumers can obtain the keys needed to decrypt it. This design eliminates reliance on intermediate parties (e.g., data storage, firewalls, or routers) to enforce access control. To automate key management, NAC uses context provided by NDN names, enabling the straightforward description of fine-grained data access controls suitable for a variety of applications. A security solution will get used only if it is easy to use; NAC's use of NDN names intends to streamline real-world application use.

Our primary research consideration is the exploration of NDN naming conventions to achieve the above concept. Below we present the basic NAC design (Section 2), followed by its extension to support attribute-based encryption [3], NAC-ABE. 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 [5] and integrated into NDN-CCL library suite [4]. An initial prototype of NAC-ABE extension is implemented as a separate C++11 library [7].

2 NAME-BASED ACCESS CONTROL (NAC)

NDN's hierarchical namespaces can be used to convey rich contextual information useful for data access control. For instance, one may name all the read access control keys under a prefix "`/example/_READ`", and the write access control keys under a prefix "`/example/_WRITE`". One can also support fine granularity in access control through a hierarchically structured namespace, e.g., "`/example/sub_space/_READ`", "`/example/sub_space/x/_WRITE`". NAC consists of a system model and specific naming conventions like these.

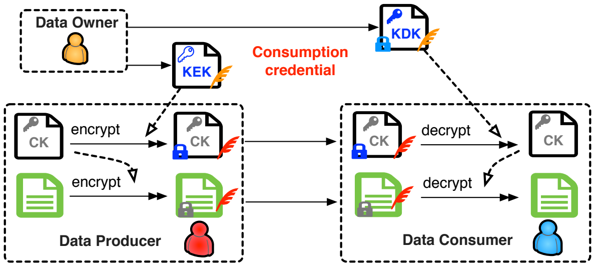

Figure 1: Major system components in NAC

NAC System Model. Figure 1 shows the overall system structure for NAC, which consists of three types of entities: data owner, data

ICN '17, September 26–28, 2017, Berlin, Germany

ACM ISBN 978-1-4503-5122-5/17/09…$15.00
https://doi.org/10.1145/3125719.3132102

producer, and data consumer.[1] The data owner provides two types of credentials: *production credential* and the *consumption credential*. A production credential authorizes a legit producer to produce and sign data under a given namespace N, which is not covered here. 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 the content key *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 the component “ENCRYPTED-BY” is a special tag. 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, one also learns the name of the encryption key.

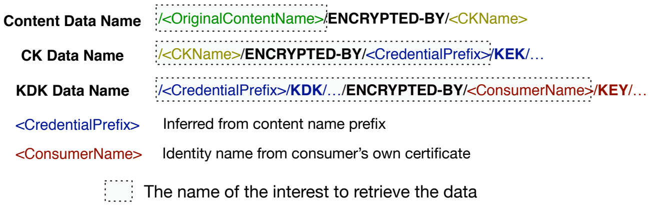
Content Data Name
/<OriginalContentName>/ENCRYPTED-BY/<CKName>
CK Data Name
/<CKName>/ENCRYPTED-BY/<CredentialPrefix>/KEK/…
KDK Data Name
/<CredentialPrefix>/KDK/…/ENCRYPTED-BY/<ConsumerName>/KEY/…
<CredentialPrefix>
Inferred from content name prefix
<ConsumerName>
Identity name from consumer’s own certificate
The name of the interest to retrieve the data

Figure 2: Naming conventions of NAC

Naming conventions can be designed based on the granularity needed in a given application. Here, we show an example implemented in our codebase. In addition to the hierarchical namespace introduced above, two timestamp components are added to the data and key names, which are used to limit user’s access to the content associated with a specified time period only. For instance, “/example/_READ/data/ENCRYPTED-BY/example/_READ/CK/20170713/20170714” cannot be decrypted by the *CK* whose suffix is “/20170712/20170713”. For brevity, further optimizations and details are omitted here.

3 NAC-ABE

To provide additional flexibility for access control policies, we extend NAC with ciphertext-policy attribute-based encryption (CPABE) [1, 2] to create NAC-ABE. In this extension, the data owner still retains the control over data production path, but the consumption control is delegated to an attribute authority (Figure 3). More specifically, a data producer in NAC-ABE encrypts data with the owner-defined policy keys, 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 all the attributes required by the policy in the form of a cryptographic key 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 data.

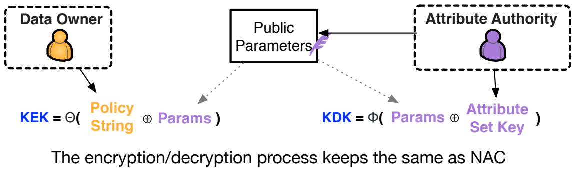
Data Owner
Public Parameters
Attribute Authority
KEK = Θ(Policy String ⊕ Params)
KDK = Φ(Params ⊕ Attribute Set Key)
The encryption/decryption process keeps the same as NAC

Figure 3: Major system components in NAC-ABE extension

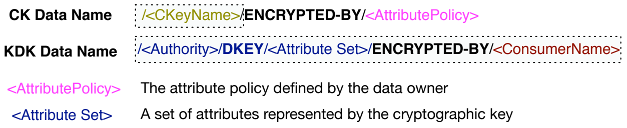
CK Data Name
/<CKeyName>/ENCRYPTED-BY/<AttributePolicy>
KDK Data Name
/<Authority>/DKEY/<Attribute Set>/ENCRYPTED-BY/<ConsumerName>
<AttributePolicy>
The attribute policy defined by the data owner
<Attribute Set>
A set of attributes represented by the cryptographic key

Figure 4: Naming conventions of NAC-ABE extension

Attribute Authority as a Level of Indirection. In the basic NAC, data owners directly manage the data access, verifying each U’s identity and encrypting KDKs for all authorized ones. This brings complexity to both data owners with a large corpus of consumers in terms of the required crypto operations, and consumers of diversely owned data in handling all the decryption keys. NAC-ABE allows data owners to simply define attributes needed to access the data, and data consumers to have sets of attributes as decryption keys. The attribute authority’s role is to properly vet the consumers and provide decryption keys to those with attested attributes.

4 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 the poster. Additionally, access rights revocation remains as one piece of the future work.

ACKNOWLEDGMENT

This work is partially supported by the National Science Foundation under awards CNS-1345318, CNS-1629922, and CNS-1719403.

REFERENCES

[1] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-policy attribute-based encryption. In *Proc. of IEEE Symposium on Security and Privacy*.
[2] Mihaela Ion, Jianqing Zhang, and Eve M. Schooler. 2013. Toward Content-centric Privacy in ICN: Attribute-based Encryption and Routing. In *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*.
[3] Amit Sahai and Brent Waters. 2005. Fuzzy identity-based encryption. In *Proc. of Conference on the Theory and Applications of Cryptographic Techniques*. 457–473.
[4] Jeff Thompson et al. 2017. NDN-CCL API. https://github.com/named-data/NDN-CCL-API. (2017).
[5] Yingdi Yu, Alexander Afanasyev, and Lixia Zhang. 2016. *Name-based access control*. Technical Report NDN-0034, Revision 2.
[6] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named Data Networking. *ACM SIGCOMM Computer Communication Review* (2014).
[7] Zhiyi Zhang and Yukai Tu. 2017. NAC-ABE Codebase. https://github.com/Zhiyi-Zhang/NAC-ABE. (2017).

[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.

[2] For simplicity, in the following description we assume that the authority can reliably verify which attributes each consumer U has and issue keys to U for the set of attributes U possesses, when requested. Other optimizations are omitted here.