IPRES: In-device Proxy Re-Encryption Service for Secure ICN

Kalika Suksomboon
KDDI Research, Inc.
ka-suksomboon@kddi-research.jp

Atsushi Tagami
KDDI Research, Inc.
tagami@kddi-research.jp

Anirban Basu
KDDI Research, Inc.
basu@kddi-research.jp

Jun Kurihara
KDDI Corporation
kurihara@ieee.org

ABSTRACT

A key paradigm of Information Centric Networking (ICN) is that the *content-based* security, privacy and access control are deployed directly in the network layer. However, there is a gap between security in the network and application layers. This creates a vulnerable space for cyber attacks from inside a device. To address this problem, we discuss and present a guideline on ICN access control. Since a semi-trusted proxy has been seen as an advantageous solution for access control and efficiency in content sharing services, we introduce the proxy in a consumer's device and propose an in-device proxy re-encryption service (IPRES) architecture for efficiency in both access control management and resource usage.

CCS CONCEPTS

• **Security and privacy** → *Security protocols*; • **Networks** → *Programming interfaces*;

KEYWORDS

Information Centric Networking, Access Control

ACM Reference Format:
Kalika Suksomboon, Atsushi Tagami, Anirban Basu, and Jun Kurihara. 2017. IPRES: In-device Proxy Re-Encryption Service for Secure ICN. In *Proceedings of ICN '17, Berlin, Germany, September 26–28, 2017,* 2 pages.
https://doi.org/10.1145/3125719.3132089

1 INTRODUCTION

A key paradigm in *Information-Centric Networking* (ICN) is that the *content-based* security, privacy and access control are deployed directly in the network layer. In IP networks, publishers verify the entities requesting accesses to the contents, whereas, in ICN, the publishers cannot always verify these since the content can be cached in any intermediate routers, and it is difficult to transfer the verification to those routers.

Recent studies, e.g., [2, 5, 6], have shown that access control has more important role in ICN security due to the unique characteristics of ICN, such as name based routing and in-network caching. However, these current layer 2/3/4 access control schemes mainly verify if "a device" is allowed to access the network. In practice, a device, such as a smartphone, normally hosts various applications, which share the device's resources. Once a device receives the keys from a content owner [6] or a trusted third party [2], the device, which is allowed to access the content, may store the keys in its cache memory. This opens the vulnerability for a malware/spyware that can eavesdrop the keys and other sensitive data.

Traditional content-based access control mechanisms manage the authorization for each application instead of a device-level access control. While they provide the most secure access, they are inefficient because of not reusing replicas in the cache and over utilizing network bandwidth due to redundant requests from different applications on the same device for the same content. Moreover, various applications may have different requirements for the access control. Thus, an interaction is required between applications and the access control in ICN.

The goal of this paper is to show an ICN-specific access control mechanism on the consumer device that can solve the aforementioned issues. To this end, we propose the in-device proxy re-encryption service (IPRES) architecture that not only efficiently handles the key and access control rights for applications, but also reduces the redundant requests.

2 IPRES

2.1 Design goals

One of the most important challenges in ICN-based content dissemination is that the publishers lose their controls on access and dissemination of content since the consumers can retrieve the copy of content from any intermediate routers and the end device. We design our access control mechanism to overcome this challenge and design an application-specific content access. The requirements of our access control include: (1) the applicability to all ICN architectures, (2) the granularity (e.g., the access right granting to each application, and the access right revocation by the publisher and the consumer), (3) the efficiency in reusing the cached content in intermediate routes and the end device.

2.2 IPRES architecture

To achieve efficiency in terms of reusing replicas and application-specific access control, a semi-trusted proxy [3] is a good candidate to handle the access control because the publisher still has the right to enforce the access control policies on its content, and the authorized applications still have their keys to access the content. On the other hand, the proxy has no right to access the content, but it is responsible for transforming content that is encrypted with

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

ACM ISBN 978-1-4503-5122-5/17/09…$15.00
https://doi.org/10.1145/3125719.3132089

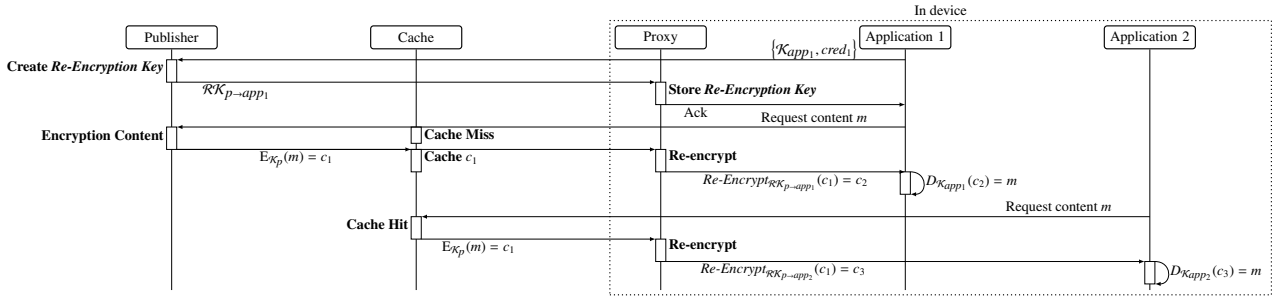

Figure 1: IPRES message sequence. $E_{key}(\cdot)$, $D_{key}(\cdot)$ and $ReEncrypt_{key}(\cdot)$ refer to the encryption, decryption and re-encryption functions with *key* respectively, and {} refers to communication over a secure channel.

the *Publisher Key* to one encrypted with the authorized *App Key*. Hence, the publisher does not lose its access control to the proxy, while the content cannot be leaked to unauthorized applications. A typical proxy re-encryption service is done by the third party in the network, e.g., [1, 5]. However, it is inefficient in interacting with various applications due to a multitude of requirements from the different applications. Hence, we introduce a proxy re-encryption service in a device and propose the IPRES architecture. Unlike the proxy re-encryption approach in [1, 5], which focuses on the access control for ICN in the network layer, our design aims to provide an access control for applications in which an authorized app is allowed to access the content while the same content can be reused in the caches. The advantages of using in-device proxy over the proxy in a storage node [1] are as follows. Firstly, a consumer can easily manage the access control via the proxy in her device. Secondly, consumers can reap the benefit of nearest-replica in edge caches and in-device cache.

The IPRES has three main building blocks:

Re-encryption: To control the right of access to the content, the publisher is responsible for creating a *Re-Encryption Key* specific to itself and an authorized app and send this key to the proxy service in a device. This key is resent once the key revocation is required.

Proxy service: The proxy service is responsible for re-encrypting the encrypted content and interacting with applications in the device. The application authenticates itself to the publisher. This constitutes the setup stage for the proxy, which receives a *Re-Encryption Key* from the publisher and uses it to re-encrypt the encrypted content for the applications. The proxy controls the semi-right of content access by deciding whether to pass the re-encrypted content to the applications or not. This allows the device user to control the permissions via the proxy service.

Interest filtering: The proxy service is placed between the applications and the device's storage so that it can filter the redundant requests for the same content from the different applications.

2.3 Message Sequence

Figure 1 shows the essentials of IPRES message sequence. In the initiation phase, Application 1 sends its *App Key* ($\mathcal{K}_{app1}$) and its credentials ($cred_1$), e.g., account name and password, to the publisher through a secure channel. The publisher authenticates the user and generates a *Re-Encryption Key* ($\mathcal{RK}_{p\to app1}$) for $\mathcal{K}_{app1}$ and *Publisher Key* ($\mathcal{K}_p$). The publisher sends $\mathcal{RK}_{p\to app1}$ to the proxy. When the proxy receives $\mathcal{RK}_{p\to app1}$, it stores the key(s) and sends an acknowledgement *Ack* to Application 1. The important point is that the proxy does not know the user's information, and it cannot decrypt the encrypted content as well. This separation achieves the access control for each application by the semi-trusted proxy. Next, Application 1 requests some *content m*, and the publisher responds by sending the encrypted content c_1, where $c_1 = E_{\mathcal{K}_p}(m)$, i.e., the encryption of the content m with the publisher's key $\mathcal{K}_p$. Upon retrieving c_1 from the publisher or an intermediate router, the proxy transforms c_1 into the encrypted content c_2 using $\mathcal{RK}_{p\to app1}$, where $ReEncrypt_{\mathcal{RK}_{p\to app1}}(c_1) = c_2 = E_{\mathcal{K}_{app1}}(m)$. The Application 1 decrypts c_2 using $\mathcal{K}_{app1}$. The encrypted content c_1 can be reused by Application 2. Assuming that the proxy has its corresponding re-encryption key, once Application 2 requests content m, the proxy responds by sending re-encrypted content c_3, where $ReEncrypt_{\mathcal{RK}_{p\to app2}}(c_1) = c_3 = E_{\mathcal{K}_{app2}}(m)$.

Note that $\mathcal{K}_{app1}$, $\mathcal{K}_p$, and $\mathcal{RK}_{p\to app1}$ can be the sets of symmetric keys [4] or the public keys [5]. The sequence after the initiation phase is common to the existing ICN-based access control schemes. IPRES is also applicable to various ICN architectures.

3 ONGOING AND FUTURE WORK

We plan to demonstrate the proof of concept of the IPRES service by integrating with NDN Forwarding Daemon (NFD). We plan to implement the IPRES in an operating system level virtualized environment, such as Docker.

Acknowledgements: This work is paper partly supported by the Horizon2020/NICT EU-JAPAN ICN2020 project, Grant Agreement No. 723014 and Contract No. 184.

REFERENCES

[1] Nikos Fotiou and George C. Polyzos. 2016. Securing Content Sharing over ICN. In *Proc. of ACM ICN*. 176–185.
[2] Jun Kurihara, Christopher Wood, and Ersin Uzun. 2015. An Encryption-Based Access Control Framework for Content-Centric Networking. In *Proc. of IFIP Networking*. 1–9.
[3] Zhiguang Qin, Hu Xiong, Shikun Wu, and Jennifer Batamuliza. 2016. A Survey of Proxy Re-Encryption for Secure Data Sharing in Cloud Computing. *IEEE Transactions on Services Computing* 13, 9 (2016), 1–1.
[4] Amril Syalim, Takashi Nishide, and Kouichi Sakurai. 2011. Realizing Proxy Re-encryption in the Symmetric World. *Informatics Engineering and Information Science* (2011), 259–274.
[5] Christopher A Wood and Ersin Uzun. 2014. Flexible End-to-End Content Security in CCN. In *Proc. of IEEE CCNC, 2014*. 858–865.
[6] Yingdi Yu, Alexander Afanasyev, and Lixia Zhang. 2015. *Name-Based Access Control*. Technical Report. NDN, Technical Report NDN-0034.