甲（Kou）
様式　４．（Form 4）

# 学　位　論　文　概　要
# Dissertation Summary

学位請求論文（Dissertation）
題名（Title）Design and Implementation of the Efficient and Secure Content Distribution Scheme in Named Data Networking
*（邦訳又は英訳）名前に基づくネットワーキングにおける効率的で安全なコンテンツ配信方式の設計と実装*

専攻（Division）: Electrical Engineering and Computer Science
学籍番号（Student ID Number）: 1824042013
氏名（Name）: Htet Htet Hlaing
主任指導教員氏名（Chief supervisor）: Masahiro Mambo

学位論文概要（Dissertation Summary）

NDN is one of the dominant candidates in the ICN architecture in which contents become the priority elements to address the current Internet architecture limitations. One of the main features of NDN is in-network caching, where contents can be cached in the intermediate routers in the network. Since the cached contents are widely available at the intermediate routers, data access control turns out to be problematic. The content producer may lose control over the contents which have been disseminated to the network. A malicious consumer can easily access the cached contents from the cache and attempt unauthorized decryption without the content producer's permission. Hence, this dissertation aims to design and implement an efficient secure content distribution solution (ES_CD) for NDN architecture, which provides content confidentiality by encryption and a limited access time for each consumer.

To deal with content security issue, encrypting contents using modern encryption schemes is one of the best ways to guarantee secure content distribution and fine-grained access control. NDN with cryptographic schemes can also lead to myriad problems, for example, key distribution, revocation, communication, and computational overhead for encrypting every content. The main problem of user revocation in NDN is that the revoked users can still access and decrypt cached data with the old keys. When revocation happens, the ciphertext needs to be updated into a new ciphertext and re-encrypted by the content producer with the new key. However, such an update leads to a high computational cost and old cached ciphertext unusable in NDN. Therefore, a new approach is needed to utilize cached contents efficiently and distribute them securely to the NDN network with the enforcement of an effective access control mechanism.

Our scheme aims 1) to disseminate the encrypted content securely only to legitimate consumers confidentially and efficiently with fast content retrieval time, 2) to enforce flexible access control on the encrypted data by securely distributing the corresponding decryption key 3) to ensure the user revocation without requiring the contact to the content producer. ES_CD can guarantee that the content producer can handle all the accesses by defining the access time while all legitimate consumers can access the requested data through a nearby router to leverage the in-network caching.

In ES_CD, sensitive contents are encrypted by the content producer using the symmetric key encryption algorithm before disseminated to the network. The corresponding symmetric key will be encrypted again by an Identity-based Proxy re-encryption scheme (IB-PRE) and integrating the time to limit the content access time for each consumer. Each content may be encrypted using the different keys to ensure that only legitimate consumers can obtain the symmetric key to decrypt the content. Both the content

producer or consumer can be active only at the start of the message exchange stage and do not need to be always online. Consumers with authenticated identities can get the decryption key to decrypt the content, and they can attempt the content decryption before the expired time. The edge router re-encrypts the key automatically after the predetermined time. The overview design architecture of our proposed ES_CD is shown in Figure.1.

If a consumer *i* wants to retrieve content, it needs to register with its identity $id_i$ to the authenticated consumer list at the content producer. As soon as the registration request arrives, the content producer first checks the consumer's $id_i$ and defines an access time *t*. Then it stores the consumer as one of the authenticated consumers and shares the list $(id_i, t)$ to the edge router. After registration is finished, the content producer replies *Ack* with the valid access time *t* to the consumer.

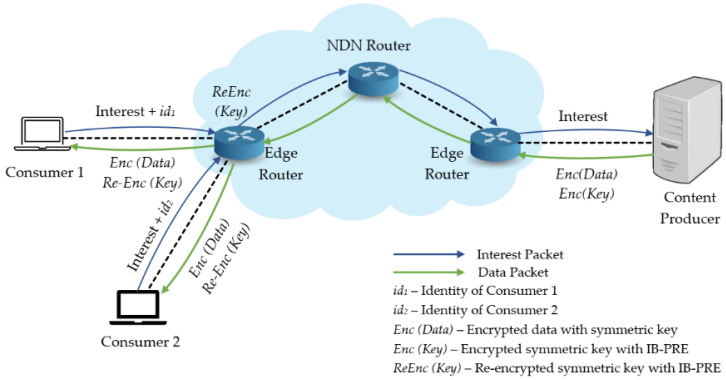


*Figure 1. Overview system architecture for our proposed ES_CD.*

Then the consumer can start the required content request to the content producer. It checks the content in the NDN router's cache; if there is no cache, it will directly request it from the content producer. When the content producer receives the request, it encrypts the requested content *content* with symmetric key *k* and produces *Enc(content)*. Then the key *k* is encrypted with IB-PRE as $c_{id_p}$ in and also generates the re-encryption key $rk_{id_p \to id_i}$ for the edge router to perform a re-encryption operation. It replies *Enc(content)*, $rk_{id_p \to id_i}$ and $c_{id_p}$ to the consumer. When the reply data packet arrives at the edge router, it extracts the re-encryption key and the first layer ciphertext of the symmetric key $c_{id_p}$ to re-encrypt $c_{id_i}$ and forwards both *Enc(content)*, $c_{id_i}$ to the consumer. At last, the consumer can decrypt the symmetric key *k* with its private key and then recover the original content *content*.

All the consumers can access both encrypted content *Enc(content)* and the first layer encrypted symmetric key $c_{id_p}$ from the cache, but the authenticated consumer can only decrypt the re-encrypted symmetric key with its private key. If the consumer requests the same content again, it can easily retrieve the encrypted content *Enc(content)* from the cache after registration. After sending interest packet, it can find the cache data at the intermediate router, but the edge router will check if the consumer's access time is still valid in. If the time is valid, the edge router will forward the corresponding re-encrypted symmetric key $c_{id_i}$ directly to the consumer. If not, it will revoke the consumer from the authenticated consumer list and the consumer have to start from the registration stage.

We set up a compact NDN environment with one content producer, one consumer, one intermediate router, and one edge router to evaluate the computational overhead of our system. The consumer requests for a 2KB content file by sending the interest request as /netflix/movie/001. Then the content producer performs content encryption, corresponding key encryption, and re-encryption key generation and appends all the keys along with the encrypted data and returns it to the consumer in a data packet as /netflix/movie/001/enc_data/enc_key/RK/. The edge route first extracts /enc_key/RK/ part from the data

packet to perform re-encryption. It re-appends to the data packet as /netflix/movie/001/enc_data/reenc_key/ and forwards it to the consumer. The consumers can get the symmetric key by decrypting the re-encrypted key, and then the requested content can be recovered. We execute the simulation 30 times by requesting the same file size by the consumer as plotted in Figure.2, which becomes linear with the number of executions.

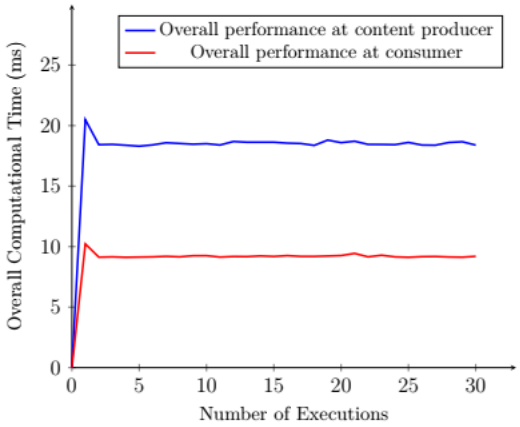


*Figure 2. Average overall computational time for retrieving 2KB content.*

We conduct another simulation with different file sizes in the same NDN environment to measure the time cost for AES encryption for the original contents at the content producer side and AES decryption time at the consumer side, respectively, as illustrated in Figure.3. The results were statistically significant that AES performance time gradually increases with respect to the content file sizes. The overall cryptographic performance time with different file sizes at the content producer and the consumer is presented in Figure.4. Compared to AES encryption and decryption process, IB-PRE operation times are not related to the content file sizes, and sometimes the overall time cost becomes lower for larger file size since we apply IB-PRE only to encrypt the symmetric key *k*, which has the same size for every content, not to directly every original content.

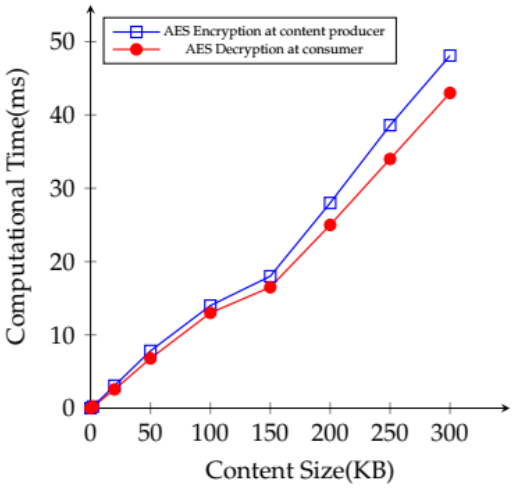


*Figure 3. Average computational time for AES operations with different file sizes.*

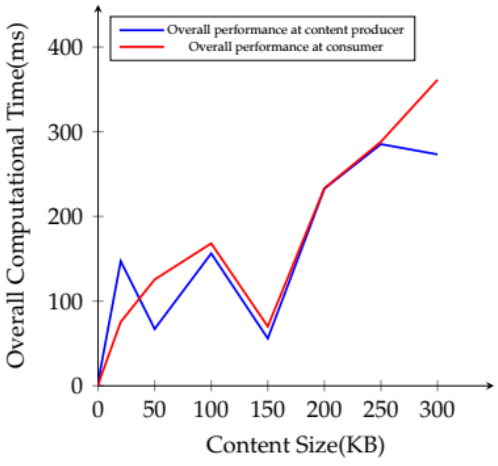


*Figure 4. Overall computational time for different file sizes*

To examine the performance analysis for network communication cost, we set up an experimental environment by constructing a different number of VMs with 2CPU and 2048MB of Memory to simulate the NDN entities. All the VMs are running on an Intel(R) Core (TM) i7-8700 CPU @ 3.20GHz and 8GB of Memory. We use the tree topology in the network simulations with one content producer, one edge router, *n* NDN routers, and *l* consumers, connected through the average bandwidth 200Mbps and 5ms delay as shown in Figure.5.

We conduct the network simulation and compare the content retrieval time for both fresh and cached content for 5 different consumers which perform successive requests, as shown in Figure.6 where each consumer sends the same content request with 2KB, and the number n of routers is 1. We run the

experiment five times and analyze the cached content retrieval for different consumers. A consumer needs to obtain both encrypted content and encrypted key directly from the content producer for the first content request cached in the router, and the content retrieval process may take more time. Figure.6 shows that the content retrieval time is reduced for the cached content by 10% to 25% on average.

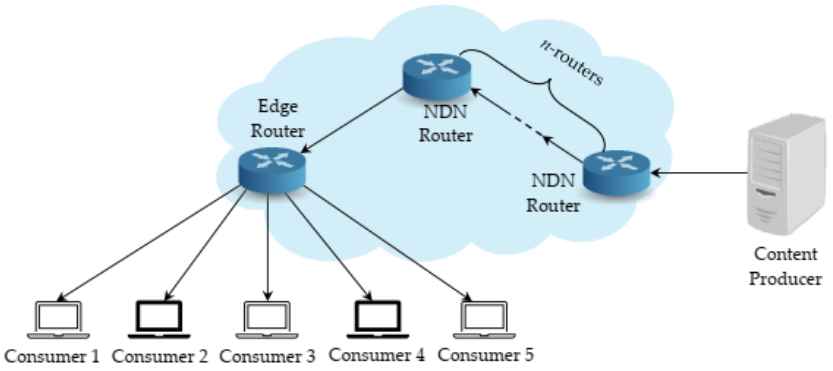


*Figure 5. Network topology for performance analyses*

Additionally, we measure the effects of adding routers in our simulation scenario to highlight the impact of communication overhead on the consumer to retrieve the content. The consumer requests 2KB content and we increase the number of routers, i.e., *n=1,2,...5*. Figure.7 shows that the content retrieval time for the consumer slightly increases with the number of routers since the consumer needs to pass through *n* numbers of routers to register at the content producer, requests and retrieves the content.

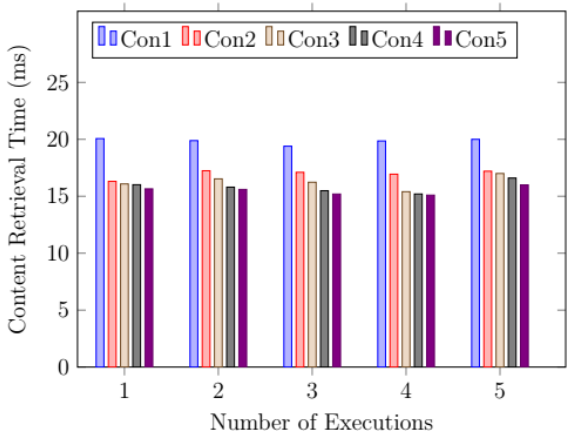


*Figure 6. Content retrieval delay for different number of consumers with different id*

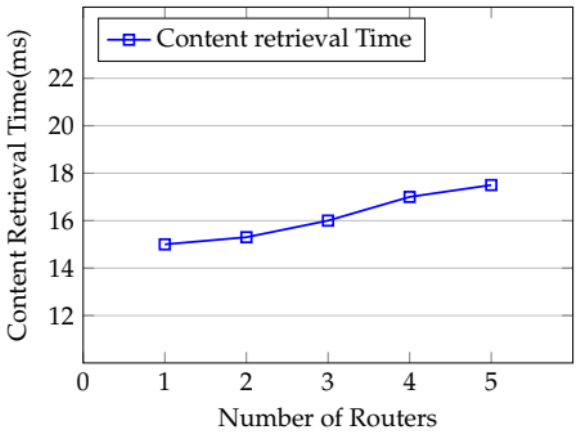


*Figure 7. Content retrieval delay for n-number of routers for retrieving 2KB content*

We compare ES_CD with the most relevant encryption-based access control scheme and prove that it offers a lower computational overhead and faster content retrieval time while protecting content confidentiality in NDN architecture through implementation. Finally, we apply our scheme in ndnSIM and analyze its performance in a large scale NDN network with a tree topology. We conduct various simulations and analyze the content retrieval time with multiple consumers and computational time for retrieving large file sizes. The simulation results show the efficiency and feasibility of ES_CD to apply in NDN architecture with a small computational and communication time.

ES_CD maintains the in-network caching ability of NDN, re-encryption and user revocation happen at the edge router so that there is no significant content retrieval delay, and it can reduce the computational overhead. Our scheme fits well with the NDN architecture with an acceptable computational and communication overhead.