ICIC Express Letters
Volume **17**, Number **3**, March **2023**
 ISSN 1881-803X
pp. **289–301**

A NOVEL ACCESS CONTROL SCHEME WITH IMMEDIATE REVOCATION OF ACCESS PRIVILEGES FOR NAMED DATA NETWORKING

Nattavut Sriwiboon and Somnuk Puangpronpitag*

Information Security and Advanced Network Research Group
Faculty of Informatics
Mahasarakham University
Khamriang Sub-District, Kantharawichai, Mahasarakham 44150, Thailand
nattavut.sri@msu.ac.th; *Corresponding author: somnuk.p@msu.ac.th

Received June 2022; accepted August 2022

Abstract. *Named Data Networking (NDN) is a new paradigm for the future Internet, aiming for efficient content delivery using in-network cache and information-centric communication. Security is built into NDN by embedding a public key signature in each data packet to enable verification of authenticity and integrity of contents. Access control is one of the most challenging issues in NDN. Several previous studies have proposed access control models over NDN. However, there are several drawbacks, particularly access revocation issues. We present a novel access control scheme to solve the problems by achieving immediate revocation. Our access control scheme enables efficient access control for NDN based on the encryption-based access control. The prototype of our scheme has been built on NDN-CXX version 0.7.1. To compare with the previous work, the evaluation has been done by algorithm analyses and emulation techniques on the CORE emulator. From the evaluation results, the proposed mechanism can provide an immediate revocation. We have also found that our access control scheme is suitable for NDN architecture, and the computational burden for immediate revocation is less than in previous proposals.*
Keywords: Named Data Networking (NDN), Access control, Immediate revocation, Information Centric Network (ICN)

1. **Introduction.** The Named Data Networking (NDN) [1,2] has considered the shortcomings of the mobility, content distribution, and security in the classical Internet architecture (Transmission Control Protocol/Internet Protocol: TCP/IP), and then proposed a new Internet architecture, shifting away from the host-centric model to the data-centric model. The data-centric model desires a security model that secures data chunks directly rather than securing hosts and channels. However, access control is one of its major challenges. Since NDN contents could be available in-network caches and be accessible by all the entities, content access control is critical. Several previous studies [3-13] have proposed the access control schemes for NDN, but there are still several drawbacks, particularly access revocation issues. So far, there have been two schemes of access control revocation, namely lazy revocation [14], and immediate revocation [15]. For the first scheme, the revoked consumers can still access all contents from in-network caches by using the existing access keys, causing security risks. For the second scheme, the previous proposals require additional nodes (proxy servers and NDN routers) to perform the revocation mechanism, and such a requirement might not be resilient in the NDN network environment.

Hence, we propose a novel access control scheme for NDN to enable efficient and immediate revocation. The revocation mechanism does not need to re-encrypt published contents. Furthermore, our proposed scheme requires no proxy and is, therefore, suitable for the NDN network. The prototype of our scheme has also been implemented in an NDN

DOI: 10.24507/icicel.17.03.289

platform. Performance evaluation has been done using emulation techniques. The results have demonstrated that the computation cost of our scheme to establish the immediate revocation is less than the previous NDN access control schemes.

The remainder of this paper is organized as follows. Section 2 explains the literature review and research motivation. In Section 3, the details of our scheme are explained. The performance evaluation is discussed in Section 4. The discussion of our scheme is provided in Section 5. In the last section, the conclusions of this work are given.

2. Literature Review.

2.1. Named data networking.

NDN started in 2010, as a promising model for the future Internet. It provides better communication availability compared to the classical Internet (TCP/IP) [16]. One of the most important deployments of NDN is expected over battlefield scenarios [1]. In such scenarios, immediate revocation of access control privileges is so crucial. We use a battlefield scenario as shown in Figure 1 to discuss in the rest of this paper. All entities are the participants in the NDN networks, and each entity has a semantically meaningful name, as follows: 1) A command center, acting as an access manager to directly control the access rights and production rights, 2) Unmanned Aerial Vehicles (UAV) A and B acting as producers, and 3) squad A and a battleship acting as consumers. NDN network nodes, including a satellite, an aircraft gateway, and a squad gateway, are NDN forwarders that distribute contents among entities.

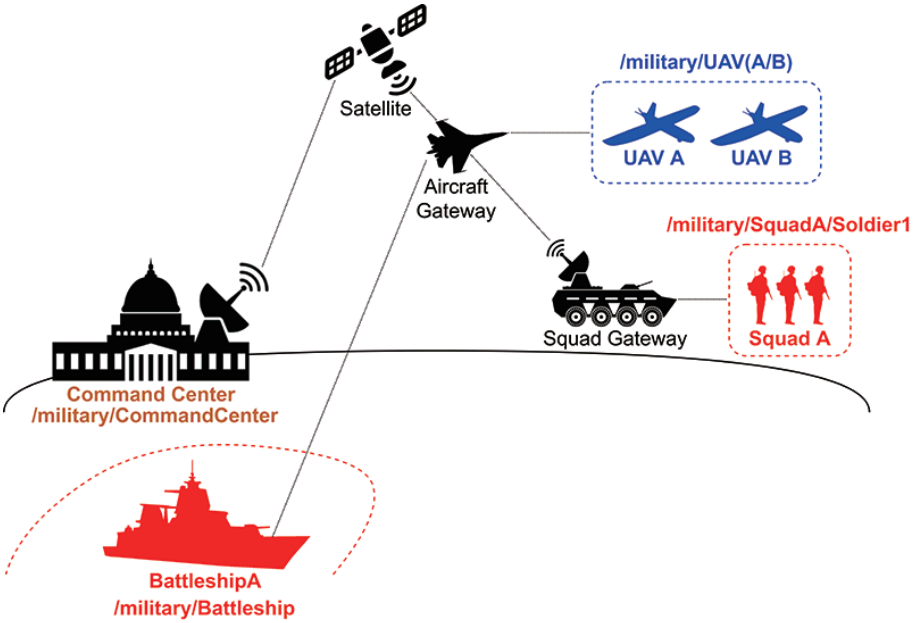

FIGURE 1. A battlefield scenario

The NDN is a prominent data-centric communication, which enables location independence that delivers packets by content names. Communication in NDN architecture relies on two types of packets: *interest* and *data packets* as shown in Figure 2. Peers participating in communication either produce or consume named data. Entities consist of UAV A as a producer, squad A as a consumer, and other network nodes. The network nodes act as NDN forwarders and third-party storage (for caching the contents).

NDN plans to reshape the Internet into a content distribution architecture. For example, UAV A needs to publish its own "info.jpg" file. This content must be split into small pieces, called chunks, to be put into *data packets*. Each *data packet* is uniquely identified by a name, according to standard naming conventions [17], e.g., "/military/UAVA/info.jpg/<chunk sequence>", where the "chunk sequence" field is the sequence of the *data*

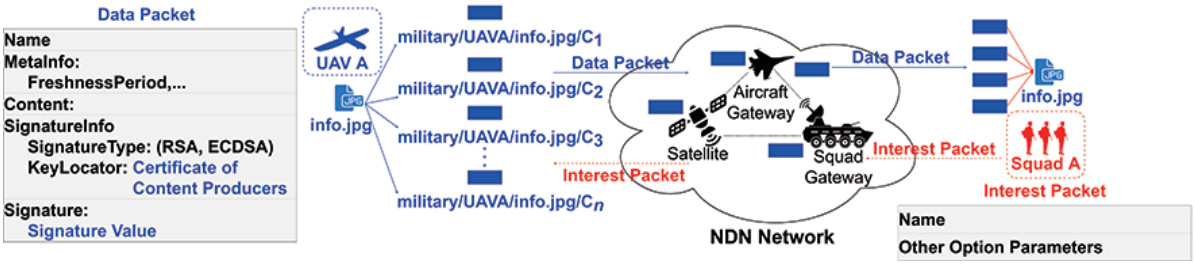

FIGURE 2. The communication paradigm of the NDN network

packet. UAV A can create a digital signature on each *data packet* at the time of production by using the NDN trust model [16]. So, the integrity and data source authentication of the *data packet* are independent of where the data are retrieved, where every *data packet* can be cached in the NDN node, and also UAV A. A soldier in squad A can receive the "info.jpg" file by defining the name as "/military/UAVA/info.jpg" into the *interest packet*, and sending this *interest packet* to the NDN node or UAV A. After obtaining the *data packets*, a soldier must fetch the certificate inside in-network caches, by using a "SignatureInfo" field, indicating the name of the producer's certificate. A soldier can use the certificate of the producer to validate the received content. In this way, the *data packet* is considered valid if all certificate chains along with the trust anchor have valid signatures.

2.2. **NDN access control.** The NDN enables content caching. So, the producers can satisfy corresponding requirements in *interest packets* irrespective of whether the requests are from authorized or unauthorized consumers. That means the content, which is available in the network is accessible by all the users. There is no access restriction applied to the available contents. Therefore, the enforcement of content access control policies becomes an issue. A promising approach is encrypting contents by a producer, and the corresponding decrypting key must be securely delivered to the authorized consumers [1]. This access control scheme is called an encryption-based access control [18,19], which is the most popular access control model over NDN.

2.3. **Related work and unresolved problems.** Most NDN access control schemes [3-13] were based on the encryption-based access control scheme. In this paper, we focus on their access revocation problems.

Several proposed schemes [3-8] were lazy revocation. We use Figure 1, as an example, to illustrate these schemes as follows. The UAV A needs to encrypt the contents individually for the soldiers in squad A and a battleship. If UAV A wants to revoke the battleship access privilege, the lazy revocation schemes need UAV A to update keys and re-encrypt the contents for each authorized consumer. Hence, lazy revocation introduces a high overhead when the revocation is needed. Moreover, UAV A cannot control the cached content in the network. The battleship can still access the contents previously published by using the existing access keys, as long as these contents are not expired.

Several proposed schemes [9-13] use proxy re-encryption to enable immediate revocation. These schemes need the producer to publish the encrypted contents to the network, and also send a delegate key (also known as a policy key) to a proxy server. The proxy server re-encrypts the encrypted contents to the authorized consumer under the policy key of the producer. These schemes can achieve immediate revocation and need no producer to re-encrypt the contents individually for each authorized consumer. The producer just re-generates the policy key and sends it to the proxy server. However, this solution requires additional nodes (proxy servers or NDN routers) to perform the revocation mechanism. Such a requirement might not be resilient in the NDN network environment. Furthermore, the computation cost can heavily occur at the proxy server because this solution needs

the proxy server to re-encrypt the contents individually for each authorized consumer, whenever the contents are accessed.

3. Our Access Control Scheme.

3.1. Design overview and assumption.

To solve the problems, mentioned in Section 2.3, our design is based on the encryption-based access control that uses cryptographic techniques to provide confidential content to authorized consumers. The notations used to explain our access control scheme are described in Table 1.

Table 1. The notations

Methods	Operations
KP	A policy key is generated by an access controller.
C_i	C_i $(i = 1, 2, 3, \ldots, n)$ represents n chunks of data.
DK_p	A decryption key is generated by a producer.
DK_i	A random decryption key i is generated by a producer.

Our scheme considers the entities, consisting of a command center, producers, NDN forwarders, and consumers. We assume the NDN trust model among all entities must be enabled via the security bootstrapping process. The producers or the consumers can send a notification to the command center whenever the revocation is needed. The design overview of our access control scheme is shown in Figure 3.

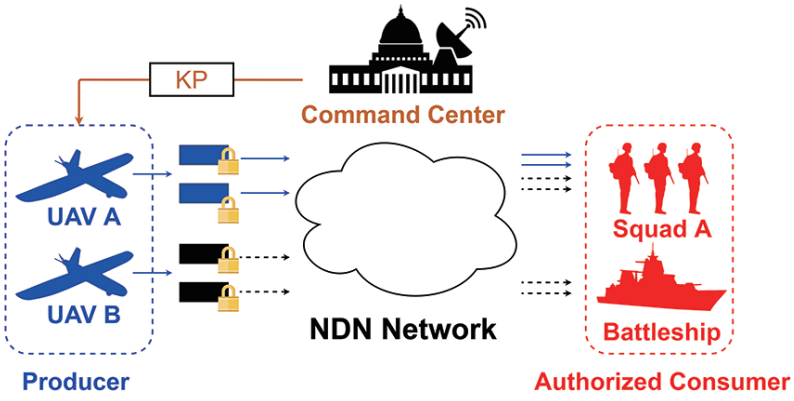

Figure 3. Our access control scheme design overview

Command Center: The command center acts as an access manager to directly control the access and production rights in the systems. It needs to generate a KP. It can use the NDN trust model to assert the KP authenticity, and also publish KP to the producers.

Producers: They can encrypt each *data packet* (chunk) under the access policy, according to its own KP. They can control the shared content by management of the DK_p and DK_i distributions, and use the NDN trust model to assert all *data packets*. In addition, they can validate the *signed interest* [20] of the authorized consumer by using the NDN trust model.

NDN Forwarders: The NDN routers can provide *data packet* storage service by the NDN communication paradigm. Hence, they can also remove the *data packets* according to the lifetimes in the "FreshnessPeriod" field in *data packets*.

Consumers: The consumers can fetch the encrypted *data packets* from the NDN routers or the producers. They can also obtain the access privilege by sending the *signed interest* to the producer.

3.2. **Our scheme.** In Figure 3, we illustrate a novel access control scheme to enable immediate revocation. The workflow of our access control scheme can be described as follows.

Policy Key Generation: The command center needs to generate the policy keys, called Key Policy (KP). It directly controls both content production and access rights. The *KP* is just like any other NDN *data packet*. It can be directly fetched through *interest packets* carrying the corresponding key names. The *KP* is named under a specific naming convention, e.g., "/military/CommandCenter/Encrypt-By/military/UAVA/KP/<key-id>", where the key-id is the unique identifier of the *KP* key. We assume that the command center can authorize each consumer by using the NDN trust model. After a consumer's certificate is authenticated, the consumer's certificate name can be used to generate the policy keys. After that, the command center generates the *KP*, which the "Content" field of *KP* carries the certificate's name of the authorized consumers. Finally, the command center uses the NDN trust model to assert the *KP* authenticity, and publish *KP* to the producers.

Encryption and Publishing: The producer can generate an *interest packet* automatically to fetch a *KP*. For example, the producer UAV A can send an *interest packet* with a name, e.g., "/military/CommandCenter/Encrypt-By/military/UAVA/KP<key-id>" to the command center. After obtaining the *KP*, the producer uses the NDN trust model to verify the authenticity of the *KP*. The producer must use the name "/military/UAVA" in the received *KP* to define the name of the encrypted content. In this way, the command center can directly control the production rights. The producer generates the DK_p and DK_i. After that, it uses a DK_p to encrypt the $C_{1,2,\ldots,n-1}$ and uses a DK_i to encrypt a C_n. The name of a C_n *data packet* must be wrapped in the $C_{1,2,\ldots,n-1}$ *data packets*. The producer uses the certificate of authorized consumers to encrypt DK_p and DK_i and defines the encrypted DK_p and DK_i in a C_n *data packet*. After that, it can use the NDN trust model to assert all *data packet* authenticity and publish $C_{1,2,\ldots,n-1}$ *data packets* to the NDN routers. The NDN routers can cache the $C_{1,2,\ldots,n-1}$. Hence, the consumer can access almost all encrypted contents from any neighbor forwarder. This paradigm enables content to be efficiently delivered to the consumers and increases mobility, and delay-tolerant networking.

Content Access: The content access process is shown in Figure 4. In step 1, the consumer can generate the *interest packet*, named "military/UAVA/info.jpg" to fetch the contents. In step 2, the NDN forwarder can send back the $C_{1,2,\ldots,n-1}$ to the consumer. If the NDN forwarder does not have $C_{1,2,\ldots,n-1}$, it can simply forward the *interest packet* toward the potential location (e.g., the producer or another NDN forwarder) of the

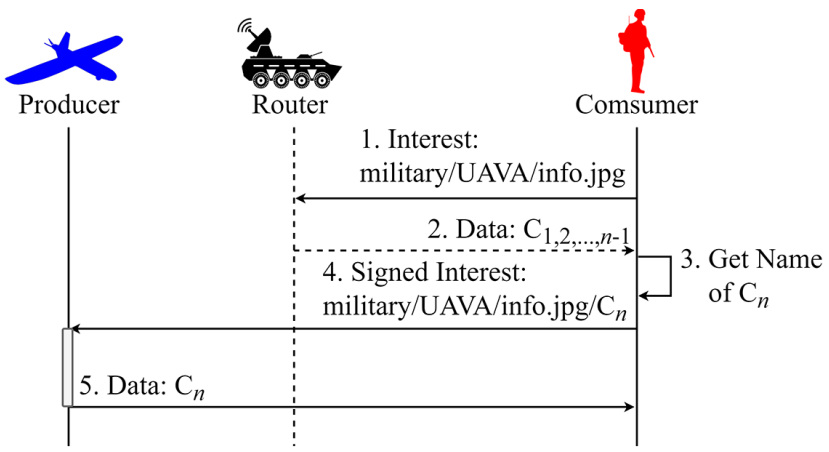

FIGURE 4. Content access process

corresponding content (according to the NDN communication model). In step 3, after obtaining $C_{1,2,\ldots,n-1}$, the consumer can learn the C_n name from the received *data packet*. In step 4, the consumer puts the C_n name in a *signed interest*. After that, it uses its private key to sign a *signed interest*. To control access rights, the consumer will send a *signed interest* to the producer. In step 5, the producer can use the certificate's name of an authorized consumer in the "Content" field of *KP* to compare with a certificate's name in a received *signed interest*. In this way, the command center can directly control the access rights. If a matching certificate name exists in *KP*, the producer can fetch a certificate by using a certificate's name in a received *signed interest* and use a received certificate of the authorized consumer to validate the *signed interest*. If the *signed interest* is authentic, the producer encrypts a C_n with DK_i and encrypts the DK_p and DK_i with the certificate of the authorized consumer. Finally, the producer generates a C_n *data packet* by wrapping it with the encrypted DK_p and DK_i and sends back the C_n *data packet* to the authorized consumer.

Decryption: After obtaining all *data packets*, the consumer uses the NDN trust model to verify their authenticity. If all *data packets* are authentic, the consumer uses its private key to decrypt the encrypted DK_p and DK_i in the C_n. To access contents, the consumer uses a DK_p to decrypt the $C_{1,2,\ldots,n-1}$ *data packets* and uses a DK_i to decrypt a C_n *data packet*.

Revocation of Access Privilege: We assume that the *KP* is regenerated whenever the revocation is needed. Our scheme allows the command center can re-publish the new *KP* by defining the new version of *KP* in the "<key-id>" field, and the producer can fetch a new version of *KP*, by sending an *interest packet* with the name of the new *KP*. In addition, our access control scheme also defines the value of "FreshnessPeriod" in the C_n *data packets* as zero, which means they are destroyed after use. For the reasons above, the producer needs to publish a new C_n to the authorized consumer according to a new policy in the *KP*. For example, as in Figure 3, if a battleship is revoked, the command center can re-publish the new *KP* by defining the certificate's name of squad A into the *KP*'s content. So, a battleship cannot access all *data packets* of the UAV B previously published by using the existing authorized key. Since the *signed interest* of the battleship does not contain the name of the certificate in the *KP* of UAV B, the *signed interest* of the battleship cannot be verified by UAV B. As a result, UAV B cannot publish a fresh C_n to a battleship. In contrast, squad A can obtain a fresh C_n *data packet* to access all *data packets* of the UAV B.

4. Performance Evaluation.

To evaluate the performance of our mechanism, the prototype was implemented into a network emulator, named Common Open Research Emulator (CORE) [21] for the experiments. We evaluate the performance in terms of key generation time, computation cost, and communication cost. Furthermore, efficiency analysis of the mechanism has been done.

4.1. Experimental setup.

Implementation: We have implemented our access control scheme as an application prototype in C++ version 7.2.0 and used the NDN-CXX version 0.7.1. This library has been also used by existing applications [22] over the NDN network.

Experimental Parameters: The CORE and network emulation techniques are deployed to evaluate the performance of our access control scheme. All entities have installed the NFD version 0.7.1 [23] for network forwarding. The following hardware is deployed for our experiments: Intel®Core i5-10400 CPU @ 29 GHz, 16GB of RAM, and the Operating System is Ubuntu 20.04 LTS. The local Network Interface Cards (NIC) of all nodes are configured to forward the *data packets* using the forwarding strategy in the NDN network.

The Ethernet links as 100 Mbps are used for experiment. Our experimental parameters are illustrated as follows.

Cryptography Algorithms: For the content keys (DK_p and DK_i), we use the 256 bits of content keys that would be used to encrypt/decrypt the *data packets* with the AES (Advanced Encryption Standard) algorithm. For other security mechanisms, we use the NDN-CXX version 0.7.1 library to operate the security mechanism, including the ECDSA of the key size 256 bits to perform the digital signature. The SHA-256 hash functions are used to compute a 256-bit hash value.

Component Names: We define the experimental parameters for the name of the command center as "/military/CommandCenter" (23 bytes), the name of the producer as "/military/UAVA" (14 bytes), the name of the *data packet* as "/military/UAVA/info.jpg" (23 bytes). To evaluate the performance of the policy key generation, we define the different certificates name of the authorized consumers as "/military/SquadA/Soldier$(1, 2, \ldots, 120)$/<key-id>" (34-37 bytes).

Payload Size: In the experiments, various typical payload sizes are used, including 5 MB, 10 MB, 15 MB, 20 MB, and 25 MB. The purpose of these various sizes is to test the parameter sensitivity in terms of payload sizes.

NDN Data Packets: We define the lifetime of $C_{1,2,\ldots,n-1}$ *data packets* to be 10 milliseconds (ms) to cache these *data packets* with 10 seconds in-network cache.

Network Nodes: The number of a producer is one node, and also one node is used for a consumer. To experiment with different numbers of NDN forwarders, various numbers of forwarders are used, including 40, 60, 80, 100, and 120 nodes.

4.2. **Experimental results to check overhead.** Our evaluation aims at a quantitative analysis of our scheme. The main point of this experiment is to check the overhead of our scheme (based on extra-encryption) in comparison to the normal NDN without encryption which cannot provide confidentiality and access control. We have performed tests on our access control scheme to observe its performance by using several metrics, including computation cost and communication cost. We use the experimental parameters shown in Section 4.1. We run each experiment 30 times. The results are averaged from the 30 runs by represented with 95% confident intervals, and also the evaluation results have been reported in ms.

Computation Cost: We observe the following overheads: 1) the extra time used in our scheme for the producer to produce and encrypt the content and for the consumer to decrypt and access the content by comparing to the time used to produce and access the content without any encryption (Figure 5); 2) the extra time to generate policy key (Figure 6). As shown in Figure 5, our scheme costs more time to produce and access the contents only 7.9% of 5 MB payload, 8.4% of 10 MB payload, 7.6% of 15 MB payload, 8.9% of 20 MB payload, and 7.8% of 25 MB payload to provide access control over NDN. This overhead is acceptable and in line with other NDN encryption-based access control schemes. In Figure 6, the extra time of the command center to generate the KP with different certificate names of the authorized consumers is demonstrated. With the different certificate names of the authorized consumers, the processing times vary. The overall KP generation time is a little overhead due to the limited size of certificate names by nature.

Communication Cost: In this paper, we defined communication cost as the time of the consumer from sending *interest packets* until receiving *data packets*. The experimental scenario is shown in Figure 7.

We evaluate communication costs in two scenarios: One with the different payload sizes, in which the number of NDN forwarders is set to 20 nodes, as shown in Figure 8; Also, the other one with the different NDN forwarder nodes, in which the payload size

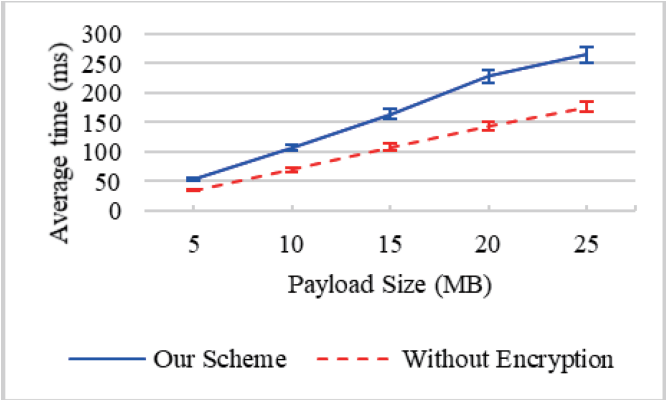

FIGURE 5. The time to generate and access content

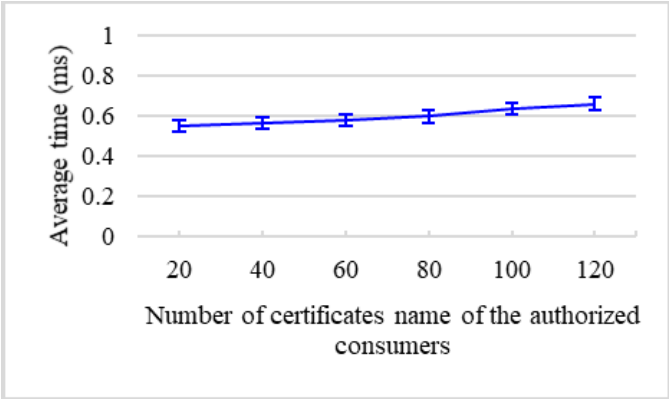

FIGURE 6. The policy key generation time

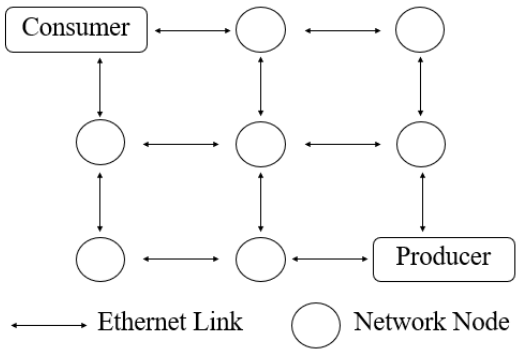

FIGURE 7. An experimental scenario

is defined as 20 MB. The communication cost is the average time of the consumer from sending *interest packets* until receiving *data packets*, as shown in Figure 9.

Figure 8 demonstrates the performance of our access control scheme in terms of communication costs with different payload sizes. We compare our scheme with the NDN content transmission without encryption (basic principles of data transmission over NDN that cannot provide access control). The experimental results have shown an acceptable overhead in terms of communication cost to provide the NDN access control. Our scheme causes the overhead of only 3.8% of 5 MB payload, 6.2% of 10 MB payload, 7.2% of 15 MB payload, 11.8% of 20 MB payload, and 9.5% of 25 MB payload.

Figure 9 demonstrates the performance of our access control scheme in terms of communication costs, with different numbers of forwarders. By comparing with the NDN

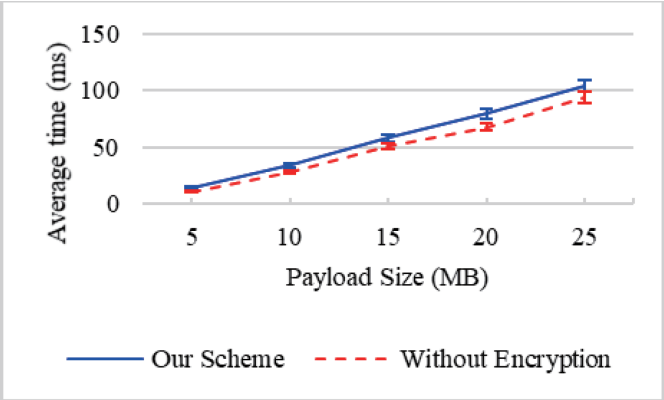

FIGURE 8. Communication costs with the different payload sizes

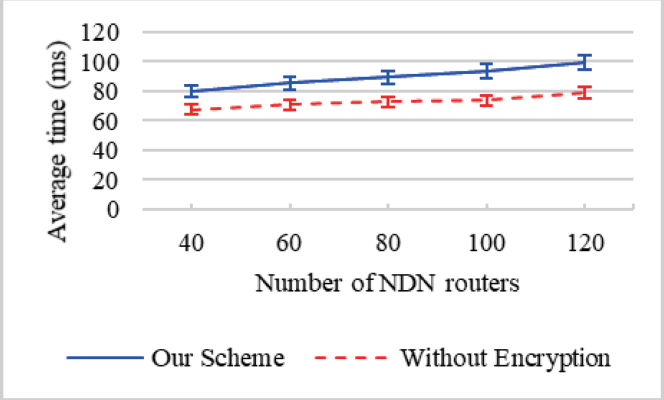

FIGURE 9. Communication costs with the different NDN routers

content transmission without encryption and access control, our scheme causes an acceptable overhead to provide the encryption-based access control. The overhead is 12.6% of 40 nodes, 14.8% of 60 nodes, 16.4% of 80 nodes, 19.7% of 100 nodes, and 20.6% of 120 nodes. The experimental results have shown that our scheme causes marginal overhead while providing data confidentiality and access control. So, we describe the performance analysis in Section 4.3.

4.3. **Efficiency comparison with other access control schemes.** To evaluate our access control scheme with other access control schemes [3-13], the performance and properties comparison are discussed as follows.

Performance Comparison: We analyze the computation burden of our access control in performing the revocation of access privilege, by comparing it to Yu et al. [5] and Wu et al. [11]. These previous studies have been picked up because there are clear details of the mechanisms, proposed in their papers. Our scheme and Wu et al.'s scheme [11] are immediate revocations while Yu et al.'s scheme [5] is lazy revocations. We define that the system has n authorized consumers, m producers, p proxy servers, and x *data packets*. The evaluation has been illustrated in Table 2.

In comparison with Wu et al. [11], the overall computation burden of our access control scheme is less than theirs, as shown in Table 2. In terms of the access manager, the revocation of access privilege is needed. Their scheme needs mnp to regenerate the new keys to m producers, p proxy servers, and n authorized consumers. However, our scheme needs only m to regenerate a new KP for m producers. In terms of the producer, their scheme needs m to update the new policy, and x to re-encrypt the contents under the new policy. Yet, our scheme needs m to update the new policy, and n to regenerate a C_n

TABLE 2. Performance comparison

Scheme	Computation burden			
	Access manager	**Producer**	**NDN forwarder**	**Consumer**
Yu et al. [5]	mn	$m+xn$	0	n
Wu et al. [11]	mnp	$m+x$	$p+nx$	n
Our mechanism	m	$m+n$	0	0

data packet for n authorized consumers. In terms of the NDN forwarders, their scheme needs $p+nx$ to update new proxy keys p, and re-encrypt x *data packets* for n authorized consumers. Our scheme has no burden due to relying only on the basic process of NDN to forward contents. In terms of the consumer, their scheme needs n to update a new access key. Yet, our scheme has no overhead due to using only the basic process of the NDN security to perform the trust model.

In comparison with Yu et al. [5], the overall computation burden of our access control scheme is less than theirs. In terms of the access manager, their scheme needs mn to regenerate the new key pairs to m producers, and n authorized consumers. In terms of the producer, their scheme needs m to update the new policy, xn to re-encrypt x contents individually for each n consumer. The overhead at NDN forwarders between ours and theirs is the same (0). In terms of the consumer, their scheme needs n to update a new access key. Yet, our scheme has no overhead due to using only the basic process of the NDN security to perform the trust model.

Properties Comparison: We discussthe properties of our revocation mechanism, comparing it against existing access control schemes. Our mechanism can enable the revocation mechanism for NDN to better solve the problems of the previous access control schemes, as shown in Table 3. The schemes [3-8] are lazy revocation by using the key update and reencryption of the contents individually for each consumer. This could incur security risks. The revoked consumer can decrypt encrypted contents, as long as the encrypted contents are cached in the NDN routers by using an existing decryption key. In contrast, our scheme enables immediate revocation. By using a fresh C_n *data packet*, the revoked consumers impossible to decrypt encrypted contents by using an existing decryption key. The schemes [9-13] are immediate revocation, similar to ours. However, they use the additional nodes (proxy servers and NDN forwarders) to perform the access control mechanism and enable immediate revocation. So, their schemes are not suitable

TABLE 3. Comparison of our access control scheme with previous studies

Mechanism	**Additional node**	**Revocation mechanism**
Hamdane et al. [3]	No Need	Lazy Revocation
Kurihara et al. [4]	No Need	Lazy Revocation
Yu et al. [5]	No Need	Lazy Revocation
Feng and Guo [6]	No Need	Lazy Revocation
Zhang et al. [7]	No Need	Lazy Revocation
Bernardini et al. [8]	Need	Lazy Revocation
Silva and Zorzo [9]	Need	Immediate Revocation
Wu et al. [10]	Need	Immediate Revocation
Wu et al. [11]	Need	Immediate Revocation
Liu et al. [12]	Need	Immediate Revocation
Jiang et al. [13]	Need	Immediate Revocation
Our scheme	No Need	Immediate Revocation

for NDN architecture that enables independent caching, in which the topology is dynamic changing like the mobile node.

Unlike the previous access control schemes, ours enables the access control scheme with immediate revocation of access privileges. Our access control scheme requires no additional nodes. It is suitable to be deployed in the NDN paradigm that enables independence from the data containers or communication channels.

5. Discussion.

Effective Access Control: Our access control scheme enables efficient access control for NDN based on the encryption-based access control. A producer can encrypt the contents at the time of production, and also it can control the shared content by management of the corresponding decryption key distributions.

Effective Consumer Revocation: By using a fresh C_n *data packet*, our scheme achieves immediate revocation. A fresh C_n *data packet* must be also regenerated by the producer under the new *KP*. In this way, the revoked consumer is impossible to access the encrypted contents in the network caches, because its *signed interest* is not authentic, and they cannot receive a fresh C_n *data packet* from the producer.

Performance and Resource Consumption: From Table 2, the evaluation indicates that previous access control schemes can cause higher overhead if they want to perform the revocation of access privilege. Wu et al.'s scheme [11] needs to explicitly use higher overhead to perform the access control mechanism by using proxy re-encryption. For example, as shown in Figure 3, the battleship and squad A can access UAV B's content. If the battleship is revoked, the access manager needs to regenerate the new keys under the new policy and publish them to UAV B, proxy servers, and also squad A. The *data packets* must be re-encrypted by UAV B, and published to the proxy servers. If squad A requests the encrypted content, the proxy servers can use the new keys to re-encrypt all content of UAV B for squad A. So, the computation cost can occur in the access manager, the proxy server, and the producer. In contrast, to revoke the battleship, our scheme needs the access manager to regenerate just *KP* under the new policy. The producer just controls the access right through a new C_n *data packet*. The authorized consumers can access almost all of the encrypted content from the NDN router without being re-encrypted.

In order to perform the revocation of access privileges, Yu et al.'s scheme [5] needs the access manager to regenerate new key pairs for each producer multiply the authorized consumers, and needs the producer to re-encrypt *data packets* individually for each consumer. So, their scheme causes a higher overhead to encrypt the contents individually for each consumer. In contrast, our scheme needs the access manager to regenerate one *KP* for the producers and needs the producer to re-encrypt only a C_n for the authorized consumers. Almost all of the encrypted content can be accessed from any neighbor router.

Efficient NDN Architecture Usage: The communication paradigm of the NDN network [24] allows the contents to be cached in the NDN routes. Our scheme needs no producer to encrypt and sign the contents individually for each consumer. The producer can control the access right through a C_n *data packet*. In contrast, the schemes [3-8] use lazy revocation by using key updates and re-encryption of the contents for each revocation. For the reasons above, their schemes require the producer to encrypt and sign individually for each consumer. This is very wasteful of the network cache storage. So, their schemes may not be completely benefited from the NDN architecture, especially caching. Moreover, our access control scheme needs no additional nodes (proxy servers and NDN routers) to perform the access control mechanism. Hence, our scheme is better than the proposed solution of [9-13] in terms of suitability for the NDN architecture and enables independence from the data containers or communication channels. In Figure 9, we evaluate the performance with the different number of forwarders. An overall comparison operation of our scheme is a little more than the normal content transmission. As the

NDN router nodes increase, the average communication cost has increased by only 3.6%. The above reasons can be explained. By using the in-network caches, our scheme can achieve the delay-tolerant, and also allow efficient recovery of the contents from losses.

6. **Conclusions.** Access control is very significant for security issues. However, the revocation problem is still a significant challenge in the access control model over NDN. The contents can be cached in several nodes. So, several previous studies have proposed access control schemes. However, their solutions are inefficient to prevent the revoked consumers from accessing all contents from in-network caches by using the existing access keys. So, this paper has evaluated the mechanism of previous work. After that, we have proposed a novel access control scheme to enable immediate revocation, which is suitable for the NDN network. We have done the performance evaluation by prototyping and experimenting using the CORE network emulator. The efficiency analysis has also been done. The evaluation results have shown that our scheme can provide immediate revocation with a lower overall computation burden, and supports the suitability to the NDN network.

For future work, we plan to enhance our scheme by using attribute-based encryption that could enhance the scalability of the scheme.

Acknowledgments. This paper was financially supported by Mahasarakham University (Grant year 2019), and also partly supported by the Newton Mobility Grant (No: NI160138) from the UK's Official Development Assistance. We are also grateful to Prof. Karim Djemame for his support, during a few months of collaboration in Leeds (UK).

REFERENCES

[1] D. K. Smetters and V. Jacobson, *Securing Network Content*, PARC Technical Report, 2009.
[2] L. Zhang et al., Named data networking, *ACM SIGCOMM Computer Communication Review*, vol.44, no.3, pp.66-73, 2014.
[3] B. Hamdane, A. Serhrouchni and S. Fatmi, Access control enforcement in named data networking, *Proc. of International Conference for Internet Technology and Secured Transactions (ICITST)*, pp.576-581, 2013.
[4] J. Kurihara, E. Uzun and C. Wood, An encryption-based access control framework for content-centric networking, *Proc. of the IFIP Networking*, France, 2015.
[5] Y. Yu, A. Afanasyev and L. Zhang, *Name-Based Access Control*, Technical Report NDN-0034 Revision 2, 2016.
[6] T. Feng and J. Guo, A new access control system based on CP-ABE in named data networking, *International Journal of Network Security*, vol.20, no.4, pp.710-720, 2018.
[7] Z. Zhang, Y. Yu, S. Ramani, A. Afanasyev and L. Zhang, NAC: Automating access control via named data, *Proc. of the IEEE Military Communications Conference (MILCOM)*, Los Angeles, CA, USA, pp.626-633, 2018.
[8] C. Bernardini, S. Marchal, M. Asghar and B. Crispo, PrivICN: Privacy-preserving content retrieval in information-centric networking, *Computer Networks*, vol.149, pp.13-28, 2019.
[9] R. Silva and S. Zorzo, An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges, *Proc. of the Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, USA, pp.128-133, 2015.
[10] Z. Wu, E. Xu, L. Liu and M. Yue, CHTDS: A CP-ABE access control scheme based on hash table and data segmentation in NDN, *Proc. of IEEE International Conference on Trust, Security and Privacy in Computing and Communications/IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, Rotorua, New Zealand, pp.843-848, 2019.
[11] Z. Wu, Y. Zhang and E. Xu, Multi-authority revocable access control method based on CP-ABE in NDN, *Future Internet*, vol.12, no.15, pp.1-13, 2020.
[12] N. Liu, S. Gao and N. Hou, CDAC: A collaborative data access control scheme in named data networking, *Proc. of International Conference on Hot Information-Centric Networking (HotICN)*, pp.44-49, 2019.

[13] S. Jiang, J. Liu, L. Wang, Y. Zhou and Y. Fang, ESAC: An efficient and secure access control scheme in vehicular named data networking, *IEEE Trans. Vehicular Technology*, vol.69, no.9, pp.10252-10263, 2020.

[14] M. Backes, C. Cachin and A. Oprea, *Secure Key-Updating for Lazy Revocation*, Technical Report RZ 3627, IBM Research, 2005.

[15] M. Blaze, G. Bleumer and M. Strauss, Divertible protocols and atomic proxy cryptography, *Proc. of EUROCRYPT*, New York, USA, pp.127-144, 1998.

[16] Y. Yu, A. Afanasyev and D. Clark, Schematizing trust in named data networking, *Proc. of ACM Conference on Information-Centric Networking (ICN)*, San Francisco, CA, USA, pp.1-10, 2015.

[17] NDN-Project-Team, *NDN Technical Memo: Naming Conventions*, Technical Report NDN-0022, 2014.

[18] U. Hengartner and P. Steenkiste, Exploiting hierarchical identity-based encryption for access control to pervasive computing information, *Proc. of the International Conference on Security and Privacy for Emerging Areas in Communications Networks*, Washington, D.C., USA, pp.384-396, 2005.

[19] S. Jahid, P. Mittal and N. Borisov, EASiER: Encryption-based access control in social networks with efficient revocation, *Proc. of the ACM Symposium on Information, Computer and Communications Security*, Hong Kong, China, pp.411-415, 2011.

[20] FIU, *Signed Interest*, https://named-data.net/doc/NDN-packet-spec/current/signed-interest.html/, Accessed in March 2020.

[21] J. Ahrenholz, Comparison of CORE network emulation platforms, *Proc. of the IEEE Military Communications Conference (MILCOM)*, CA, USA, pp.166-171, 2010.

[22] FIU, *ndn-cxx: NDN C++ library with eXperimental eXtensions 0.7.1 Documentation*, https://named-data.net/doc/ndn-cxx/current/INSTALL.html/, Accessed in January 2020.

[23] FIU, *Named Data Networking Forwarding Daemon (NFD) 0.7.1 Documentation*, https://named-data.net/doc/NFD/current/INSTALL.html/, Accessed in January 2020.

[24] Y. Liu, X. Zeng, R. Han and P. Sun, Toward information-centric networking receiver-driven transmission mechanism over wireless local area network: Implementation and optimization, *Journal of Innovative Computing, Information and Control*, vol.17, no.3, pp.853-871, 2021.