DACPI: A Decentralized Access Control Protocol for Information Centric Networking

Eslam G. AbdAllah, Mohammad Zulkernine, and Hossam S. Hassanein
School of Computing, Queen's University
Kingston, ON, Canada
Email: {eslam, mzulker, hossam}@cs.queensu.ca

***Abstract*—Current Internet architecture is becoming inadequate for new requirements of highly scalable and efficient distribution of contents. Information Centric Networking (ICN) is one of the alternatives for the Next Generation Internet (NGI), which focuses mainly on contents. In-network caching is one of the major attributes of ICN, which allows contents to be cached in any ICN node. Any user can access ICN contents from different distributed locations. This attribute maximizes the problem of unauthorized access to ICN contents. In this paper, we propose a Decentralized Access Control Protocol for ICN architectures (DACPI). In this protocol, fewer public messages are needed for access control enforcement between ICN subscribers and ICN nodes than the existing access control protocols. DACPI depends on ICN self-certifying naming scheme. We perform security analysis on DACPI for the following attacks: man-in-the-middle, forward security, replay attacks, integrity, and privacy violations. According to the security analysis, DACPI prevents unauthorized access to ICN contents with fewer messages passed.**

I. Introduction

Internet is changing from Internet of hosts to Internet of things, Internet of media, Internet of service, and Internet of people. These new Internets require highly scalable and efficient contents distribution. Information Centric Networking (ICN) is a new communication paradigm for these new Internets. Different architectures have been proposed for ICN such as Data Oriented Network Architecture (DONA), Network of Information (NetInf), Named Data Networking (NDN), and Publish Subscribe Internet Technology (PURSUIT). All ICN architectures have some commonly shared concepts, which can be classified as follows: information object, naming, routing, caching, security, and application programming interface [1].

In-network caching is a major attribute of ICN, which allows any node to cache any content. In the current Internet architectures, contents are stored at specific points, which simplifies the access control mechanisms. In ICN, subscribers can access contents from different locations. This new attribute makes the access control mechanism in ICN much more complicated. Contents in ICN can be classified into open access contents and restricted access contents [2]. We are concerned about restricted access contents that must be accessed by legitimate users only.

Existing access control schemes can not be applied directly to ICN architectures because of the following three reasons: ICN supports the in-network caching, ICN requests do not have any user identification information, and ICN does not depend on IP addresses [3]–[11]. There are many malicious requests for ICN architectures related to naming, routing, caching and unauthorized access [12]–[14]. In this paper, we are concerned with unauthorized access attacks. Access control mechanisms can be classified as centralized, decentralized, and encryption-based mechanisms. In centralized access control, there are extra entities such as authentication servers or key generation and distribution centers. These entities are responsible for evaluating ICN users against access control policies for ICN contents. In decentralized access control mechanisms, ICN subscribers and nodes work together for mutually authenticating each other and ensure that legitimate users access legitimate contents. In the latter case, ICN publishers are also included in this authentication process. In the encryption-based mechanisms, access control is satisfied by encrypting ICN contents or requests or both.

In this paper, we propose a Decentralized Access Control Protocol for ICN (DACPI) between ICN subscribers and nodes based on ICN self-certifying naming scheme. ICN self-certifying naming is unique, persistent, not limited to any organization and makes it easier for integrity checking. In this protocol, we use the following cryptographic techniques: public key infrastructure, key exchange using Diffie-Hellman, hashing, and random number generations [15]. The main objective of this protocol is to allow only legitimate users to access legitimate contents. We perform security analysis based on the following: man-in-the middle, forward security, replay attacks, content or request modifications, and privacy violations to ICN users.

Our contributions in this paper can be summarized as follows: We prevent unauthorized access attacks in ICN using fewer number of public messages with respect to the existing solutions by proposing the DACPI. We show the effectiveness of the DACPI by performing security analysis.

The remainder of this paper is organized as follows. Section II shows generic access control schemes and presents the related work in ICN access control literature. Section III presents the proposed decentralized access control protocol (DACPI). In Section IV, we analyse the security for DACPI and compare it with the existing access control protocols. Finally Section V presents our conclusions.

II. Background and Related work

This section covers the background and related work for access control schemes in ICN. We start with generic central-

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

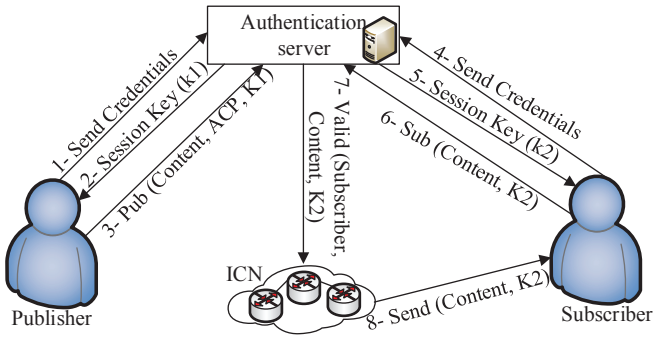

Fig. 1: Generic centralized access control scheme in ICN.

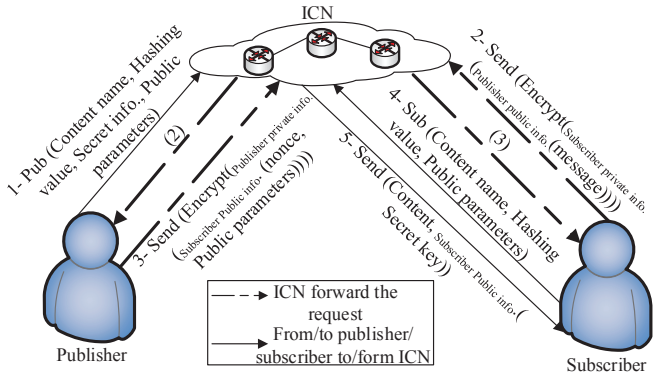

Fig. 2: Generic decentralized access control scheme in ICN.

ized and decentralized access control schemes in ICN. In the last subsection, we discuss the existing specific access control schemes in ICN in the context of our proposed protocol.

A. Centralized access control schemes

In the ICN centralized access control schemes [3], [4], there are extra entities responsible for achieving access control services. All ICN requests go through a central authority that accepts or rejects the requests. This scheme simplifies the access control, however, no access can be granted if the central authority fails. There are some variations to generic centralized schemes. One scheme may use more than one type of extra servers. Some of these extra servers are authentication servers that are responsible for managing access control policies and the other servers are responsible for managing ICN nodes to send ICN contents to legitimate users. Another scheme may use key distribution center to distribute secret keys for ICN users and depend on ICN name resolution systems for managing contents and access control policies.

Figure 1 shows a generic scheme for a centralized access control scheme in ICN. In this figure, the authentication server indicates the extra entity responsible for access control policy or key distribution.

First, a publisher sends his credentials to the authentication server and the authentication server replies with a session key, if the publisher is a legitimate user. Then the publisher sends a publication message with the ICN content name, access control policy, and the session key. A subscriber starts also by sending his credentials and the authentication server replies with another session key, if the person is a legitimate user. Then the subscriber sends a subscription message using the ICN content name and the session key. The authentication server evaluates the subscriber's request against the access control policy attached with the ICN content. If the subscriber is an authenticated user for this content, the authentication server sends a message to the ICN to forward the content from the best available copy to this subscriber.

In this generic scheme, the secret keys (k_1, k_2) can also be used in different ways to encrypt and decrypt ICN contents and requests and the keys are distributed in a secure way.

B. Decentralized access control schemes

In the ICN decentralized access control schemes [5], there is no need for extra entities or architecture modifications. ICN users and nodes collaborate together for mutual authentication. The access control responsibilities are distributed across many locations. The decision of accepting or rejecting an ICN request can be taken by ICN nodes themselves.

This approach is more stable because no single point of failure exists. If one ICN node fails, the request can be redirected to any other node. There are also some variations that can be used for access control. Many security techniques can be used such as public key infrastructure, identity-based cryptography, hashing, shared secrets generation and distribution, to name a few.

Figure 2 depicts a generic decentralized access control scheme. A publisher sends a message with the content name, hashing value, secret information, and public parameters. Hashing value is used for integrity checking, while secret and public parameters are used for confidentiality. A subscriber sends an encrypted message with the content name that achieves both authenticity and confidentiality, and ICN forwards the request to the responsible publisher. Then the publisher replies with an encrypted message that also achieves both authenticity and confidentiality with a nonce and public parameters to the subscriber. The subscriber afterwards sends a request with the content name, hashing value, and public parameters to ICN nodes. ICN evaluates the request with respect to the received publication and then sends the content back to the subscriber. Finally the subscriber also evaluates the content and secret information to be sure that everything is authenticated.

C. Specific ICN access control schemes

The existing work controls the access to ICN contents by different ways such as centralized [3], [4], decentralized [5], and encryption-based access control schemes [6]–[11].

For the centralized access control techniques, Fotiou et al. [3] propose an access control scheme that evaluates subscriber's requests against access control policies. This scheme uses extra entities such as relaying party and access control

provider. Relaying party is responsible for content distribution to legitimate users. Access control provider allows publishers to create contents with access control policy and evaluate subscriber's requests. Aiash et al. [4] present an authentication and authorization scheme for NetInf ICN architecture. This scheme uses an extra entity named trusted ticket granting, which is responsible for key generation and distribution. This scheme also uses ID-based cryptographic scheme for securing the messages. The name resolution system in NetInf is responsible for authentication for ICN users and content publication and subscription.

For the decentralized access control schemes, Wang et al. [5] propose a generic session-based authentication scheme that can be applied to any ICN architecture. In this scheme [5], the authors propose two names for each ICN content, one is public and the other is a secure name that is known only by legitimate users. The scheme depends on symmetric keys that need to be generated and distributed in a secure manner.

For the encryption-based solution, Kurihara et al. [6] present an access control framework for content centric networking (CCN-AC). This framework depends on symmetric, asymmetric, and nonce keys to encrypt and decrypt the manifest and content objects. The framework uses extra entities such as key manager and access policy manager and require modifications to the content metadata. Ghali et al. [7] describe an interest-based access control for content centric networking. This scheme hides content names from unauthorized subscribers by using encryption-based name obfuscation. Misra et al. [8] use a broadcast encryption technique for a framework to deliver contents to legitimate users. ICN contents are encrypted with symmetric keys and these keys are distributed to ICN users by session keys. Ion et al. [9] propose attribute-based encryption to achieve access control and can also be used in ICN routing. Wood et al. [10] propose encryption-based technique to personalize ICN cached contents to each ICN legitimate user. Smetters et al. [11] present an access control scheme for the previous version of named data networking architecture, which depends on a group-based access control scheme.

Our proposed access control scheme (DACPI) falls in the decentralized access control mechanism category. DACPI dsoes not require extra entities or architecture modifications like the centralized schemes or content modifications similar to the encryption-based techniques.

III. Proposed protocol (DACPI)

In this section, we present the proposed protocol (DACPI) in detail. We start with the ICN reference model that we have used in our protocol. In subsection III.B, we present DACPI specifications. In subsection III.C, we show DACPI internal steps that are carried out by subscribers and ICN nodes.

A. ICN reference model

This ICN reference model consists of ICN routers, ICN legitimate users, and attackers. ICN routers contain both routing and caching capabilities. ICN users are classified into

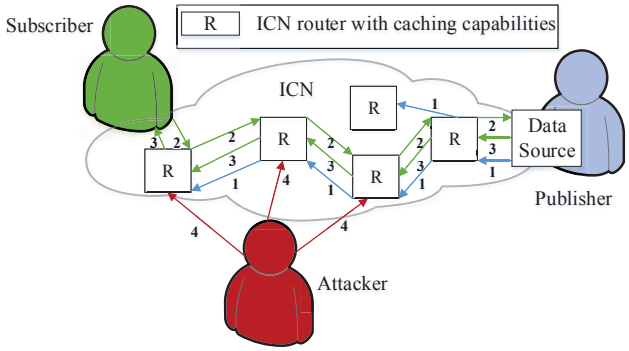

Fig. 3: ICN reference model: 1- Publish message. 2- Subscribe message. 3- Delivery path for subscriber. 4- Subscribe messages from an attacker.

publishers and subscribers. ICN publishers send publication messages to ICN about their contents. ICN subscribers can send subscription messages for ICN contents. Figure 3 depicts a generic model for the ICN architecture that we have used as a reference model. In this model, we are focused on unauthorized access attackers. These attackers try to get restricted access contents that should not be accessed by them. In the current Internet architectures, contents are stored in specific servers and caching points, which simplifies the access control mechanisms on these storage points. In ICN, contents are stored in multiple caching points and the attackers try to access these distributed copies, which simplifies unauthorized attacks and complicates access control mechanisms in ICN architectures.

Routing techniques in ICN can be classified into two approaches: name-based and name resolution routing. In name-based routing, the request is routed directly based on content name and the content is returned back using the reverse path. In name resolution routing, the content name is resolved first to one or more addresses, then the request is forwarded to one of these addresses based on shortest path routing. DACPI can be applied in both routing techniques. The reference model supports the in-network caching, which is one of the main components in ICN that provides three main attributes as follows: applicable to all contents delivered by any protocol; applicable to all publishers; available to all network nodes. The security in ICN should be applied to the content itself and integrated within the architecture. In ICN, publication and subscription functions are carried out in an asynchronous manner. Both functions use a content name as the main parameter.

To support DACPI, ICN routers should maintain the following extra tasks: compare between two hashing values from a subscriber and a publisher; calculate a shared secret key. Additionally, the metadata associated with the content contains more details than the ICN normal metadata. Our protocol's metadata contains hashing value, nonces and other secret and public parameters.

TABLE I: Notations

Notation	Definition
n_1, n_2	Nonces generated by a publisher and subscriber, respectively
p	Public large prime number used as the base for key exchange
a	Primitve root used for key exchange
PU_{pub}, PR_{pub}	Public and private key pairs for a publisher
PU_{sub}, PR_{sub}	Public and private key pairs for a subscriber
x, X	Publisher's secret and public numbers
y, Y	Subscriber's secret and public numbers
S	Shared secret key

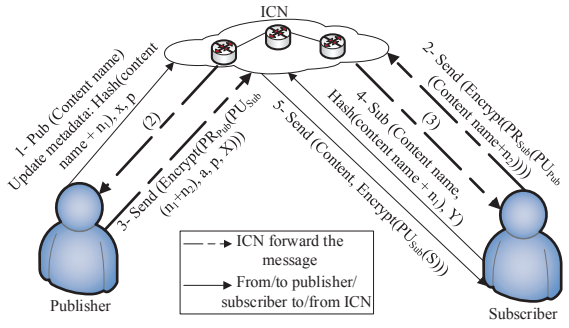

Fig. 4: Proposed decentralized access control protocol in ICN (DACPI)

B. DACPI specifications

In this subsection, we describe the proposed decentralized access control protocol (DACPI). The protocol depends on public key infrastructure and uses Diffie-Hellman for key exchange. Some of the public messages are encrypted using public key of the receiver and then using the private key of the sender to achieve authenticity and confidentiality. In DACPI, we depend on self-certifying naming scheme and use its security features. In ICN self-certifying naming scheme, names consist of two parts of the form *P:L* and metadata associated with each content. The two parts represent a unique ICN content name. The metadata is used for authenticity and integrity checking. The first part (*P*) is the cryptographic hash value of the owner's public key. The second part (*L*) is a content label assigned by the owner. Metadata contains the full public key and digital digest signed by the owner. A subscriber requests the content with both parts *P* and *L* and receives the content, publisher's public key and signed hash value. The subscriber then checks that the hash value of the received public key is the same as the sent hash value to insure that the content is received from an authenticated publisher. The subscriber also decrypts the received signed hash value and compares it with the hash value of the content to ensure that the integrity is satisfied. Table I shows the notations used in the proposed protocol. In the following paragraphs, we describe DACPI's steps in detail.

Step 1: *Pub (Content name)*

Update metadata: Hash(content name + n_1), x, p

As shown in Figure 4, in step 1, the publisher sends a publication message with the content name as in the self-certifying naming scheme that consists of the *P:L*. The publisher updates the metadata attached with the content with the following information: hashing value of content and nonce (n_1), secret information (x) and public parameter (p) which should be a large prime number. The content name is the only published part. The remaining parameters in the metadata are used in later steps for access control purposes.

Step 2: *Send (Encrypt(PR_{Sub}(PU_{Pub}(content name + n_2))))*

The subscriber sends a message with the content name and nonce (n_2) encrypted using publisher's public key and subscriber's private key. ICN forwards the message to the publisher.

Step 3: *Send (Encrypt(PR_{Pub}(PU_{Sub} (n_1 + n_2), a, p, X)))*

The publisher decrypts the message using subscriber's public key and publisher's private key and extracts nonce (n_2). In step 3, the publisher replies with an encrypted message with the subscriber's public key and then publisher's private key. The reply message contains two nonces (n_1) and (n_2), primitive root (a), public parameters (p) and (X). ICN forwards the reply message to the subscriber.

The subscriber decrypts the message using publisher's public key and using subscriber's private key and extracts nonce (n_1). The subscriber also calculates public parameter (Y) and secret key (S), as shown in the following internal steps:

Calculate hash(content name + n_1)

$Y = a^y \bmod p$

$S = X^y \bmod p$

Step 4: *Sub (Content name, Hash(content name + n_1), Y)*

The subscriber sends a subscription message to ICN with the content name, hash value of content and nonce (n_1), and public parameter (Y). ICN then evaluates the subscription message and calculates secret key (S), as shown in the following internal steps:

Compare two hash values from publisher and subscriber

$S = Y^x \bmod p$

Step 5: *Send (Content, Encrypt(PU_{Sub}(S)))*

ICN sends the content back to the subscriber with encrypted secret key (S), if the subscriber is an authenticated user. The subscriber evaluates the reply from ICN and accepts the content, if it comes from an authenticated node. We explore how DACPI design overcomes the security attacks in detail in subsection IV.A.

C. Internal steps of DACPI

In DACPI, there are internal extra steps that are carried out by ICN subscribers and nodes. These extra steps are used for evaluating the messages between subscribers and nodes. Figure 5 shows the sequence diagram of DACPI between ICN publishers, subscribers, and nodes. In this figure, we focus on the extra internal steps, while other steps are explained in detail in Figure 4. In step i1, the subscriber calculates

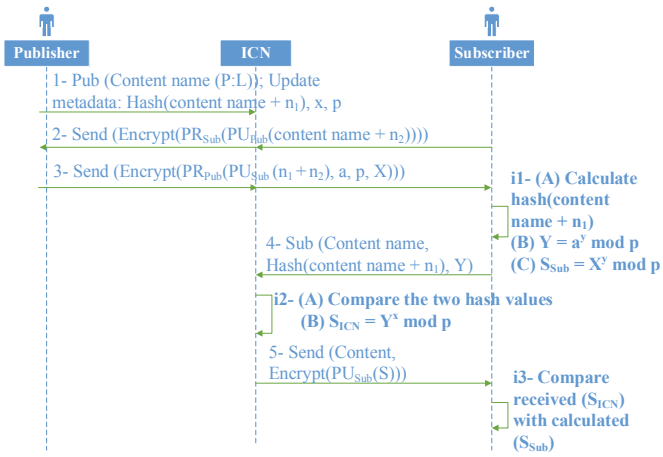

Fig. 5: Sequence diagram of DACPI

the hash value of the content and nonce (n_1), and the public parameter (Y) to send these values in the subsequent public message. The subscriber also calculates the shared secret key (S_{sub}). In step i2, the ICN node compares the two hash values and also calculates the shared secret key (S_{ICN}). If the two hash values are the same, then ICN node sends the content to this legitimate subscriber. In step i3, the subscriber compares between the received and his calculated secret key. If the two keys are the same and self-certifying naming verification is valid, then the subscriber ensures that the content is coming from an authenticated node.

IV. Analysis of DACPI

In this section, we analyse DACPI. We perform security analysis and comparison with the existing access control mechanisms.

A. Security analysis

In the proposed protocol, using only the public parameters transmitted in the public messages is not sufficient for an attacker to calculate the secret values. In this subsection, we provide a comprehensive list of attacks that may happen in ICN access control schemes. For DACPI, we explore the following: man-in-the-middle, forward security, and replay attacks. Additionally, for ICN architectures, we explore the following: ICN content or request modifications and privacy violations for ICN users.

Man-in-the-middle attacks. An attacker plays the classical man-in-the middle attack to impersonate an authenticated subscriber or publisher or ICN node. To impersonate the subscriber, an attacker needs to forward the subscription messages to persuade an ICN publisher and node that he/she is an authenticated subscriber. These messages will be invalid, because they must be encrypted using subscriber's private key and different nonce (n_2) is used with every subscription request. Additionally, an attacker also can impersonate an authenticated publisher or ICN node to persuade ICN subscriber that he/she is an authenticated publisher or ICN node. This will also be invalid, as the attacker must know the publisher's private key and published nonce (n_1) with the ICN content and initial secret information (x) to correctly communicate with the subscriber. Using the available public parameters, these secret keys and nonces cannot be easily detected by the attacker.

Forward security. This is a feature that ensures that past communications between ICN nodes and subscribers are secure, even if one communication is compromised. An attacker cannot conclude the previous requests, even if the attacker succeeds to attack one communication. Forward security is guaranteed in this protocol, as our mutual authentication depends on using private keys of publishers (PR_{pub}) and subscribers (PR_{sub}), shared secret key (S), and nonces (n_1, n_2). Private and shared secret keys depend on large prime numbers that cannot be easily broken. New nonces are used for each content publication and subscription request. In DACPI, there is no way to calculate the previous nonces.

Replay attacks. An attacker can store the public messages used in the protocol and replay these messages in a later stage to access ICN contents. The attacker will not be able to form valid messages as the protocol uses a different nonce with each request, and hence the attack is unsuccessful.

ICN content or request modifications. The access control protocol depends on self-certifying naming scheme, which ensures the data integrity for ICN contents. An attacker can try to modify the public messages during transmission. This attack cannot be done under our protocol for two reasons. First, only authenticated subscriber, who has private key (PR_{sub}), nonce (n_1) and shared secret key (S) can form a valid message. Second, only authenticated node, which has secret information (x) and nonce (n_1) can form a valid message. If the ICN subscriber or node finds any change in the requested contents or the public messages, then no more communication messages will be transferred.

Privacy violations of ICN users. An attacker monitors the public messages to gain private information about content popularities and who requested which content. All parameters in the public messages introduced by our protocol are encrypted using public key infrastructure or converted by one-way hashing function. These public parameters cannot be used to determine the secret values. At this point, we are not concerned about privacy issues from the ICN architectures, we focus here on privacy issues from the perspective of the proposed protocol.

B. Comparison with the existing ICN access control protocols

In this subsection, we show a comparison of the DACPI and the related centralized and decentralized access control protocols as shown in Table II based on the following criteria: number of public authentication messages, number of extra entities, number of secret keys, number of ICN names for each content, and required security techniques.

The table shows that DACPI has the minimum number of public messages and does not require any additional modifications to the architecture. In [3], eight public messages are

TABLE II: Comparison between DACPI and existing centralized and decentralized access control protocols

	Fotiou et al. [3]	Aiash et al. [4]	SAC [5]	DACPI
Number of public authentication messages	Eight for both subscription and publication	Four for subscription and four for publication	Six for subscription and five for publication	Five for both subscription and publication
Number of extra entities	Two (access control provider and relaying party)	One (trusted ticket granting)	Zero	Zero
Number of secret keys	Two secret keys one for publisher and one for subscriber	Three secret keys and each publisher and NRS has public and private keys	Three symmetric keys and each content provider has public and private keys	One secret key and each user has public and private keys
Number of ICN names for each content	One	One	Two (public and secure name for each content)	One
Required security techniques	No extra techniques are required	Public key infrastructure and hashing	Public key infrastructure, hashing, and random number generation	Public key infrastructure, hashing, Diffie-Hellman, and random number generation

needed to enforce the access control for both publication and subscription functions. In [4], a total of eight messages are needed for access control distributed as four messages for publication and four for subscription. In [3], two extra entities are needed, while in [4], one extra entity is needed for access control purposes. In SAC [5], a total of eleven messages are needed for publication and subscription. SAC does not need extra entities to ICN architectures.

For the required secret information, DACPI uses the same or fewer secret keys than the other protocols. In DACPI, we need one shared secret key and each user has a public and a private key. All the three access control schemes use only one name for each ICN content except the decentralized scheme [5] that uses two names for each content. The secure name in [5] will be known to authenticated users only. DACPI uses extra cryptographic techniques with respect to the other protocols.

V. Conclusion

ICN shifts the Internet paradigm from host oriented to content oriented. Contents are highly distributed and cached everywhere in ICN architectures. ICN subscribers access contents from different locations, which complicates the access control for restricted contents in ICN architectures. In this paper, we propose a decentralized access control protocol for ICN (DACPI) to achieve access control security service to ICN contents. DACPI is one of the decentralized access control schemes, which does not require new entities to ICN architectures. Access control decisions in DACPI are taken by ICN nodes and subscribers without central authority.

The protocol successfully prevents man-in-the-middle attacks, forward security, replay attacks, integrity, and privacy attacks related to the access control process. The main idea is that an attacker cannot retrieve the secret keys based on public parameters used in the proposed protocol. The attacker cannot form a valid authentication message without knowing the secret keys. The protocol efficiently minimizes the number of public messages for the access control process. With the light communication overhead used in DACPI, the access control security level is remarkably increased.

Acknowledgment

This research has been funded in part by Mitacs Canada and Irdeto Canada Corporation.

References

[1] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking", *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, 2015, pp. 1441-1454.

[2] C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, G. Xylomenos, C. Ververidis, V. Siris, and N. Fotiou, "A survey of information-centric networking research", *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, 2013, pp. 1024-1049.

[3] N. Fotiou, G. F. Giannis, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures", *The Second Edition of the ICN Workshop on Information-Centric Networking*, ACM, 2012, pp. 85-90.

[4] M. Aiash and J. Loo, "An integrated authentication and authorization approach for the network of information architecture", *Journal of Network and Computer Applications*, vol. 50, 2014, pp. 73-79.

[5] Y. Wang, M. Xu, Z. Feng, Q. Li, and Q. Li, "Session-based access control in information-centric networks: design and analyses", *IEEE Performance Computing and Communications Conf. (IPCCC)*, Austin, TX, Dec. 2014, pp. 1-8.

[6] J. Kurihara, E. Uzun, and C. A. Wood, "An encryption-based access control framework for content-centric networking", *IFIP*, 2015.

[7] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, "Interest-based access control for content centric networks (extended version)", *arXiv:1505.06258v1*, 2015.

[8] S. Misra, R. Tourani, and N. E. Majd, "Secure content delivery in information centric networks: design, implementation, and analyses", *ICN*, Hong Kong, China, 2013, pp. 73-78.

[9] M. Ion, J. Zhang, M. Schuchard, and E. M. Schooler, "Toward content-centric privacy in ICN: attribute-based encryption and routing", *ASIA CCS13*, Hangzhou, China, 2013, pp. 513-514.

[10] C. A. Wood, and E. Uzun, "Flexible end-to-end content security in CCN", *IEEE CCNC*, Las Vegas, NV, Jan. 2014, pp. 858-865.

[11] D. K. Smetters, P. Golle, and J. D., Thornton, "CCNx access control specications", *PARC*, Tech. Rep, 2010.

[12] M. Vahlenkamp, M. Whlisch, and T. C. Schmidt, "Backscatter from the data plane - threats to stability and security in information-centric networking", *Computer Networks*, vol. 57, no. 16, 2013, pp. 3192-3206.

[13] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "Countermeasures for mitigating ICN routing related DDoS attacks", *The 10th Int. Conf. on Security and Privacy in Communication Networks (Securecomm14)*, Beijing, China, 2014, pp. 84-92.

[14] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "Detection and prevention of malicious requests in ICN routing and caching", *The 13th IEEE Int. Conf. on Dependable, Autonomic, and Secure Computing (DASC-2015)*, Liverpool, UK, 2015, pp. 1741-1748.

[15] W. Stallings, "Cryptography and network security: principles and practice", Sixth Edition, Prentice Hall, 2013.