Received: 2 May 2018 | Revised: 19 June 2018 | Accepted: 1 July 2018 | Published on: 2 August 2018

DOI: 10.1002/spy2.33

ORIGINAL ARTICLE

WILEY

Preventing unauthorized access in information centric networking

Eslam G. AbdAllah | **Mohammad Zulkernine** | **Hossam S. Hassanein**

School of Computing, Queen's University, Kingston, Ontario, Canada

Correspondence
Eslam G. AbdAllah, School of Computing, Queen's University, Kingston, Ontario, Canada.
Email: eslam@cs.queensu.ca

The increasing traffic volume and new requirements of highly scalable and efficient distribution of contents exceed the capabilities of the current Internet architecture. Information centric networking (ICN) is a new communication paradigm for the next generation internet (NGI), which focuses mainly on contents. ICN has in-network caching capability, which enables any node to cache any content coming from any publisher. ICN subscribers are able to access contents from different distributed locations. This capability maximizes the problem of unauthorized access to ICN contents. In this paper, we propose a decentralized elliptic curve-based access control (ECAC) protocol for ICN architectures. In this protocol, fewer public messages are needed for access control enforcement between ICN subscribers and ICN nodes than the existing access control protocols. ECAC protocol depends on ICN self-certifying naming scheme. We perform security analysis on ECAC for the following attacks: man-in-the-middle, forward security, replay attacks, integrity, and privacy violations. We also evaluate communication, computational, and storage overhead for performance analysis to ECAC. Based on our results that are obtained under various scenarios, ECAC efficiently prevents unauthorized access to ICN contents.

KEYWORDS

access control, elliptic curve cryptography, information centric networking, unauthorized access attacks

1 | INTRODUCTION

The 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 one of the alternatives for these new Internets. The number of objects in ICN is expected to be several orders of magnitude higher than the number of nodes in current Internet architectures. According to Cisco Visual Networking Index, there will be almost 4.1 billion Internet users and 26.3 billion network devices and connections globally, the average fixed broadband connection speed will increase to 47.7 Mbps, and IP video will represent 82% of all traffic by 2020.[38]

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).[39] All ICN architectures have some commonly shared concepts, which can be classified as follows: information object, naming, routing, caching, security, and application programming interface.[40–46]

In-network caching is a major attribute of ICN, which allows any node to cache any content. This attribute is one of the major differences between ICN and non-ICN architectures. In the current Internet architectures, contents are stored at specific points, which simplifies the access control mechanisms. Network security administrator can deploy their security modules

More discussions about related work and generic centralized and decentralized access control mechanisms[1–37] can be found in our earlier conference paper.[4]

in these specific points. In ICN, subscribers can access contents from different locations.[47,48] In-network caching attribute makes the access control security service in ICN much more complicated. Contents in ICN can be classified into open access contents and restricted access contents. We are concerned about restricted access contents that must be accessed by legitimate users only.

Existing access control mechanisms cannot be applied directly to ICN architectures because of the following three reasons: ICN supports in-network caching, ICN requests do not have any user identification information, and ICN does not depend on IP addresses. Also, existing access control mechanisms fail to address the following attributes of a "good" access control mechanism: (1) reduce unnecessary communication overhead, (2) eliminate modifications to ICN architectures, (3) minimize the exchange of secret keys, (4) do not require large number of extra operations for access control, (5) prevent access control attacks, (6) preserve the privacy of ICN users, (7) ensure the integrity of the retrieved content, and (8) can be applied with different ICN naming schemes (self-certifying and hierarchical).[1–12]

There are many malicious requests for ICN architectures related to naming, routing, caching, and unauthorized access.[13–23] 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 (DACPIs), 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 based on elliptic curve cryptography (ECC; elliptic curve-based access control, ECAC). To the best of our knowledge, this paper presents the first use of ECC for access control in ICN architectures. EC can be used to achieve encryption and decryption goal and key exchange goal. EC offers equal security for a smaller key size compared to the well-known RSA technique, hence reducing processing overhead. For example key size of 112 bits in EC is equivalent to key size of 512 bits in RSA; key size of 512 bits in EC is equivalent to key size of 15 360 bits in RSA. Also, ECC is considered in standardization efforts such as the IEEE P1363 standard for public key cryptography.[24] ECAC uses the following techniques: EC, hashing, and random number generations. The main objective of ECAC protocol is to allow only legitimate users to access legitimate contents. We perform security analysis for ECAC based on the following attacks: man-in-the middle, forward security, replay attacks, content or request modifications, and privacy violations of ICN users. We do performance analysis based on communication overhead, computational overhead, and storage requirements.

Our contributions in this paper can be summarized as follows:

- We propose a decentralized access control protocol that does not require extra entities or architecture modifications for ICN.
- We prevent unauthorized access attacks in ICN using fewer number of public messages with respect to the existing solutions.
- We show the effectiveness of ECAC by performing security and performance analysis and comparing ECAC with the related protocols.

We evaluate ECAC in various scenarios and under different request rates and number of attackers with respect to the number of legitimate users. We measure ICN performance metrics in the presence of and without ECAC, a representative of DACPI,[4] and a representative of centralized access control mechanism (Access Control Enforcement Delegation, ACED).[1] Our results show that the ECAC requires less overhead than the DACPI in terms of ICN request access time delay and almost achieves results similar to the centralized ACED. However, ACED requires more public access control messages and architecture modifications.

The remainder of this paper is organized as follows. Section 2 presents the proposed decentralized ECAC. This section shows ECAC specifications, internal steps, and ECAC analysis. In Section 3, we show the impact of our proposed protocol on ICN performance. Section 4 compares ECAC protocol with the existing access control protocols. Finally, Section 5 summarizes the paper and presents the future work.

2 | PROPOSED PROTOCOL (ECAC)

In this section, we present the proposed protocol (ECAC) based on ECC. We start with ECAC specifications, then we present the internal steps of the proposed protocol. We also perform security analysis and compare between our proposed protocol and the attributes of good access control mechanisms. In the last subsection, we do performance analysis for ECAC.

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

TABLE 1 ECAC notations

Notation	Definition
n_1, n_2	Nonces generated by a publisher and subscriber, respectively
q, a, b, G	Elliptic curve public parameters
$P_{\text{Pub}}, n_{\text{Pub}}$	Public and private key pairs for a publisher
$\text{PubP}_\text{m}, \text{PubC}_\text{m}$	Plaintext and cipher text of a publisher
P_{sub}, n_{sub}	Public and private key pairs for a subscriber
$\text{SubP}_\text{m}, \text{SubC}_\text{m}$	Plaintext and cipher text of a subscriber
k_1, k_2	Random numbers generated by a publisher and subscriber, respectively
x, X	Publisher's secret and public keys for key exchange
y, Y	Subscriber's secret and public keys for key exchange
S	Shared key

Our goal is to prevent unauthorized access attacks in ICN, achieving the following explicit design goals:

- Access control: We design ECAC to prevent attackers from accessing ICN contents and allow only legitimate users to access ICN-restricted access contents.
- Minimal change to ICN architectures: Our proposed protocol does not require any extra entities or changes to ICN architectures. We only add authentication messages and cryptographic operations for access control.
- ICN utilization: Any access control protocol requires extra overhead to the architecture. We propose ECAC protocol that minimizes the required overhead. We use different metrics such as number of public messages and request access time delay to measure the required overhead based on our solution.

Also, in designing ECAC, we make the following assumptions:

- The backbone network is secure: We assume that our protocol will be applied in ICN edge routers because these edge routers are accessible by users. The aim of this assumption is to minimize extra authentication messages. In this case, required authentication messages are needed between ICN users and edge routers.
- The ICN naming scheme is self-certifying: The proposed protocol is based on ICN self-certifying naming scheme, which is a promising technique in ICN. ICN architectures such as DONA, NetInf, and PURSUIT are using this naming scheme. Access control protocol based on self-certifying naming does not need to check ICN content integrity and publisher authenticity because they are verified in this naming scheme.

2.1 | ECAC specifications

In this subsection, we describe the proposed decentralized access control protocol (ECAC). ECAC uses ECC for two purposes: encryption and decryption, and key exchange. ECC is an efficient cryptographic technique for public key cryptosystems. The security of ECC comes from the EC logarithm problem, which means the difficulty of calculating discrete logarithms in group of points defined over a finite field on an EC. Public messages in ECAC are encrypted using public key of the receiver and random positive integer chosen by the sender. In ECAC, we depend on self-certifying naming scheme. Table 1 shows the notations used in the ECAC. In ECAC, an ICN publisher uses EC encryption for plaintext $PubP_m$ to generate cipher text $PubC_m$ by the following equation:

$$PubC_m = (k_1 G, PubP_m + k_1 P_{sub})$$

Similarly, an ICN subscriber uses EC encryption for plaintext $SubP_m$ to generate cipher text $SubC_m$ using the following equation:

$$SubC_m = (k_2 G, SubP_m + k_2 P_{Pub})$$

The ICN publisher receives subscriber's secure message and decrypts the message to retrieve subscriber's plaintext by the following equations:

$$SubP_m = SubP_m + k_2 P_{Pub} - n_{Pub} * (k_2 G)$$
$$SubP_m = SubP_m + k_2 * (n_{Pub} G) - n_{Pub} * (k_2 G)$$

Similarly, the ICN subscriber receives publisher's secure message and decrypts the message to retrieve publisher's plaintext by the following equations:

$$PubP_m = PubP_m + k_1 P_{sub} - n_{sub} * (k_1 G)$$
$$PubP_m = PubP_m + k_1 * (n_{sub} G) - n_{sub} * (k_1 G)$$

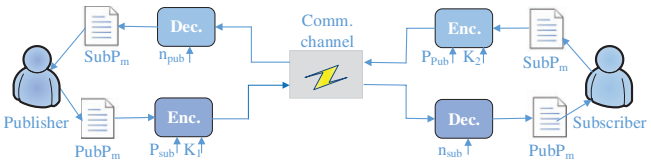

FIGURE 1 Encryption and decryption processes in ECAC: A subscriber encrypts plaintext $SubP_m$ using publisher's public key P_{Pub} and secret number k_2 and the publisher decrypts the message using the publisher's private key n_{Pub}. A publisher encrypts plaintext $PubP_m$ using subscriber's public key P_{sub} and secret number k_1 and the subscriber decrypts the message using the subscriber's private key n_{sub}. The two keys k_1 and k_2 are not transmitted between ICN publishers and subscribers

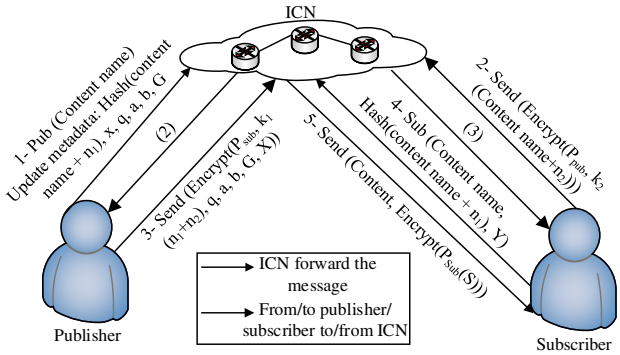

FIGURE 2 Proposed decentralized access control protocol in ICN (ECAC)

For both publisher and subscriber, EC shared key calculations can be calculated as follows:

$$S = x * Y = x * (y * G) = y * (x * G) = y * X$$

Figure 1 shows the encryption and decryption processes in ECAC. In the following paragraphs, as shown in Figure 2, we describe ECAC's steps in detail.

Step 1: *Pub (Content name).*

Update metadata: Hash(content name+n_1), x, q, a, b, G

ICN publisher sends a publication message with the content name consisting of the *P:L* parts as in self-certifying naming and metadata. The first part *(P)* is the cryptographic hash of the owner's public key. The second part *(L)* is a content label assigned by the owner. The metadata attached with the content is updated with the following information: hashing value of content and nonce (n_1), secret information (x) and EC public parameter (q, a, b, G), where q is a prime or an integer of the form 2^m and G is a point on EC whose order is a large value n such that $nG = 0$. In ECAC, EC encryption and decryption achieve message confidentiality, while shared key, hashing technique, self-certifying naming achieve content authenticity and integrity.

Step 2: *Send (Encrypt(P_{Pub},k_2[content name + n_2])).*

The subscriber encrypts a message using publisher's public key and random positive integer k_2. The value of k_2 will not be known to anybody except for the subscriber. The message itself contains the content name and nonce (n_2). ICN routers forward the message to the publisher.

Step 3: *Send (Encrypt(P_{sub},k_1 [$n_1 + n_2$], q, a, b, G, X)).*

In order to extract nonce (n_2), the publisher decrypts the message using publisher's private key without knowing the value of k_2. Then the publisher sends another message encrypted using subscriber's public key and random positive integer k_1. The value of k_1 will not be known to anybody except for the publisher. The message itself contains two nonces (n_1, n_2), EC public parameters, and publisher's public key used for key exchange. ICN routers forward the message to the subscriber.

In order to extract nonce (n_1), the subscriber decrypts the message using subscriber's private key without knowing the value of k_1. The subscriber also calculates public parameter (Y) and key (S), as shown in the following internal steps:

Calculate hash[content name + n_1]

$$Y = y * G$$

$$S = y * X$$

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

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

Compare two hash values from publisher and subscriber

$$S = x * Y$$

Step 5: *Send (Content, Encrypt(P_{sub}(S)))*

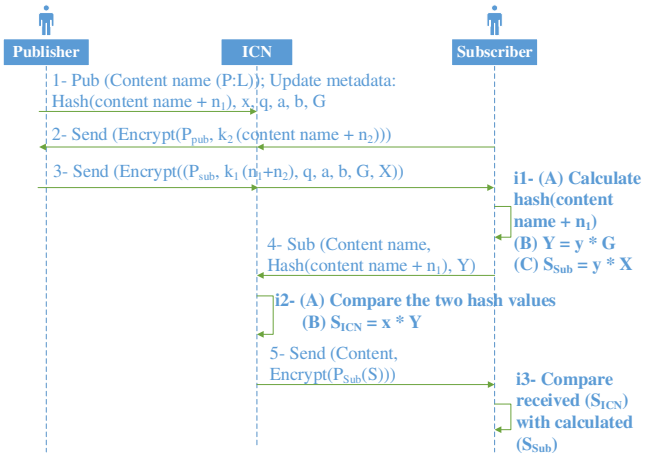

FIGURE 3 Sequence diagram of ECAC

If the subscriber is verified as an authenticated user, then the ICN edge router sends the requested content to the subscriber. This content is sent with the shared key (S) encrypted using subscriber's public key. Finally, the subscriber evaluates the reply from ICN edge router and accepts the content, if it comes from an authenticated node by performing the following verifications: First, the subscriber compares between the hash values of the sent P of ICN self-certifying naming with the received publisher's public key. This step ensures the authenticity of the received content.

$$Hash(received\ Ppub) = P$$

Second, the publisher verifies the received signature and then compares between the hash values of the received content with the calculated one. This step ensures the integrity of the received content.

$$Hash(received\ content) = Verify_{Ppub}(Sign(Hash(content)))$$

2.2 | Internal steps of ECAC

We describe the required internal steps of ECAC in order to achieve access control for restricted-access contents in ICN architectures. The purpose of these steps is to evaluate the public messages exchanged between subscribers and ICN routers and do some calculations. The sequence diagram of ECAC involving ICN publishers, subscribers, and nodes is depicted in Figure 3. The goal of the first internal step *i1* is to enable the subscriber to extract nonce n_1 to formulate the hash value of content name plus the extracted nonce. Then the subscriber calculates public number Y for key exchange purpose that will be sent later in the protocol. This public parameter is calculated using subscriber's private key that is used for key exchange and public parameter G. The subscriber afterward calculates the shared key *(S)*. The aim of the second internal step *i2* is to evaluate the subscriber's request to check whether the subscriber is an authenticated user or not. ICN edge router compares between its hash value and the received hash value from the subscriber and then calculates the shared key *(S)*. In the third internal step *i3*, the subscriber evaluates the response from ICN edge router to check its validity and authenticity by comparing the received shared key with subscriber's calculated key in addition to ICN self-certifying naming checking.

2.3 | Security analysis

Using only the public parameters transmitted in the public messages is not enough for an attacker to gain access to the restricted-access ICN contents. An attacker can be a publisher, subscriber, or man-in-the-middle attacker who performs active and passive attacks. Active attacks include impersonation, and content/request replay, and modification. Passive attackers include eavesdropping and privacy violations. An attacker uses ICN characteristics to read requests and contents during transmission. An attacker also controls some ICN nodes to forward and route contents/requests.

In this subsection, we explore a comprehensive list of attacks that may happen in ICN access control mechanisms. For ECAC, we explore the following: man-in-the-middle, forward security, and replay attacks. In addition to the following attacks for ICN architectures: ICN content or request modifications and privacy violations for ICN users.

2.3.1 | Man-in-the-middle attacks

To impersonate an ICN subscriber, an attacker needs to know subscriber's private keys and EC random number (k_2). To impersonate ICN publisher or an edge router, an attacker needs to know publisher's private keys and EC random number (k_1).

Additionally, the attacker needs to know the used nonces (n_1, n_2), which are different from each content request and shared keys. Random numbers in EC cryptography are not transmitted between different users. The attacker needs these secret information in order to form valid messages to perform unauthorized access attacks. Using the public parameters used in ECAC, the attacker cannot get these secret information. Consequently, the attacker messages will be invalid and the attacks will be unsuccessful and easily detectable. The difficulty of determining secret number (k_1, k_2) comes from the discrete logarithm problem in EC. In EC, the addition operation associates to each pair of points on EC to obey the abelian group idioms. Multiplication is defined as repeated addition, where the addition is performed over an EC. The addition between two points on the EC depends on the selection of q parameter.

$$a * k = a + a + a \ldots (k \ times)$$

Cryptanalysis involves determining k given a and $a*k$, which is the discrete logarithm problem in EC.

2.3.2 | Forward security

Using ECAC, past communications are secure, even if an attacker succeeds to get unauthorized access to one ICN content because these past communications use different nonces (n_1, n_2) and different random numbers (k_1, k_2). Also, ECAC uses EC private keys that depend on large prime numbers or an integer of the form 2^m. These keys in ECAC cannot be easily calculated using the available public information in the protocol.

2.3.3 | Replay attacks

Traditionally, an attacker stores public messages and replay them again in a later stage to persuade the other party that it is an authenticated entity. This type of attack is prevented in ECAC due to the continuous change of the used parameters with each message. Replay attacks are forms of network attack in which valid content or request transmissions are maliciously repeated. This can be done by a malicious publisher who retransmits the content or by a malicious subscriber who retransmits the requests. Also, man-in-the-middle attacker can retransmit both requests and contents. ECAC guarantees the detection and prevention of such attacks by using nonces (n_1, n_2) for each request and content response. ECAC publishers and subscribers also mutually authenticate each other by public and private parameters to make sure that each content or request is coming from the intended party.

2.3.4 | ICN content or request modifications

If an ICN subscriber or an ICN edge router finds any change in ICN content or public messages, then the communication is stopped and one party detects that the other one is not a legitimate entity. For content integrity verification, ECAC depends on ICN self-certifying scheme that does this job. For message verifications, both parties cannot form a valid message without knowing the required secret parameters. Using ECAC, content and request integrity are verified and any modifications can be easily detected.

2.3.5 | Privacy violations of ICN users

The main target of an attacker in this type of attacks is to calculate used secret keys in order to get access to private information. These private information enable the attacker to know about content popularities and requested ICN users. Using ECAC, an attacker cannot determine these secrets due to all aforementioned reasons including ECC, nonces, random numbers, and secret keys. Taking into consideration that ECAC is not targeted to prevent other ICN privacy-related attacks. We focus on privacy violations from the proposed protocol.

2.4 | Analysis of ECAC attributes

In this subsection, we evaluate ECAC with respect to the eight attributes of a good access control mechanism (see the attribute list in Section 1). The communication overhead is only extra five messages in ECAC. The existing protocols are using at least eight messages. There is no required modification in the ICN architectures for our proposed access control protocol. There is no need to add any extra entities. All the modifications are in the public messages or some extra computations done by ICN entities. This covers the reduction of unnecessary communication overhead and the elimination of modifications to ICN architecture attributes of a good access control mechanism.

The protocol is based on one shared key that needs to be exchanged in ECAC protocol. The other secret parameters (k_1, k_2) do not need to be exchanged. For ICN users, each one has a public and private key pair. In the proposed protocol, we have included hashing calculations and hashing verifications. Also, we have added encryption for transmitted messages, random

number generations, and calculations for shared key. This satisfies the minimization of the exchange of secret keys and extra operations for access control attributes of a good access control mechanism.

The protocol prevents man-in-the-middle attack, as an attacker cannot impersonate an authenticated user or ICN node because the attacker needs to know secret information that cannot be easily compromised using the public messages in ECAC. Additionally, the protocol prevents forward security and replay attacks. This achieves the fifth attribute of a good access control mechanism.

The privacy attribute has been achieved by using ECC and nonces. Information that can lead to privacy violations in ECAC are encrypted using receiver's public key in addition to the random number selected by the sender, which means that no one can decrypt it except for the intended receiver. Other ICN privacy-related issues are left as a future work. By comparing the two hash values, ICN nodes are able to check the integrity of ICN subscriber's requests. By comparing the two keys, in addition to self-certifying verifications, ICN subscribers are able to check the integrity and authenticity of ICN contents. This fulfills the sixth and seventh attributes of a good access control mechanism.

For the used ICN naming scheme, we use the self-certifying naming scheme in ECAC because it is the most popular ICN naming scheme that is used in many ICN architectures. The related access control protocols mentioned in this paper[1–3] also use self-certifying naming scheme. Hierarchal naming scheme needs additional security add-ons to achieve content authenticity and integrity.

2.5 | Performance analysis

In this subsection, we analyse communication, computational and storage overheads caused by ECAC on ICN architecture. In the next section, we experimentally evaluate how these overheads affect ICN performance.

2.5.1 | Communication overhead

In ECAC, we achieve mutual authentication between ICN edge routers and subscribers in five public messages as in DACPI. Other centralized and decentralized mechanisms require more authentication messages than our proposed protocol. In order to calculate shared key and encrypt/decrypt public messages, there are three internal steps that are not considered as communication overhead because they are not transmitted between ICN routers and subscribers. We include more parameters and encrypted information to ICN public messages, which lead to a rise in the size of these access control communication messages.

2.5.2 | Computational overhead

In ECAC, we build our access control technique based on ECC that are used for key exchange and encryption/decryption, random number generation and hashing techniques. The security level of access control in ICN is increased, using these extra techniques.

2.5.3 | Storage requirements

Although there is no need for extra entities in ECAC, there are extra required storage in ICN publishers and routers to maintain the additional parameters. For ICN publisher, an extra storage is required for storing EC public parameters and nonces. For ICN edge router, an extra storage is needed to store EC public parameters and shared key. Using this light storage overhead, the security level of access control in ICN architectures is remarkably increased.

3 | EVALUATION OF ACCESS TIME DELAY

In this section, we evaluate the impact of ECAC on ICN performance. We measure access time delay without ECAC protocol and related access control protocols and in the existence of these protocols. Request access time delay is measured for different request rates, cache sizes, and various ratios of attackers to legitimate users.

3.1 | Simulation environment

Access time delay in ICN is a major issue because of the ICN property that each request has one response and there is no response without a request. In non-ICN architectures, requests can receive many data packets. This means that security solutions are applied on each request. Our objective is to achieve access time delay results in ICN architectures close to centralized access control mechanisms. Centralized access control mechanisms add extra entities to manage access control polices to reduce

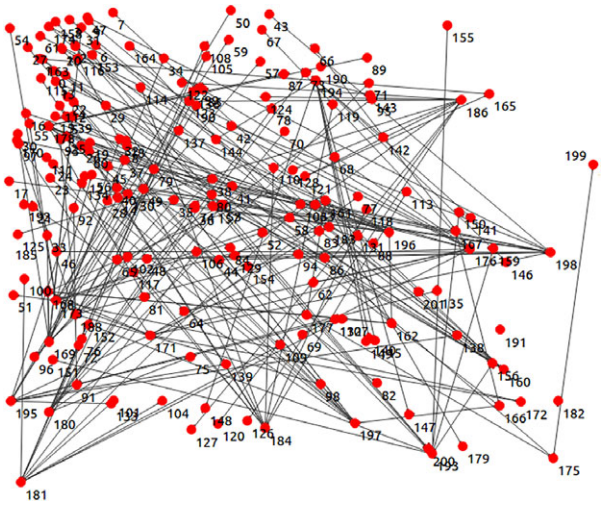

FIGURE 4 AT&T network: 150 subscriber, 10 publishers, and more than 40 routers

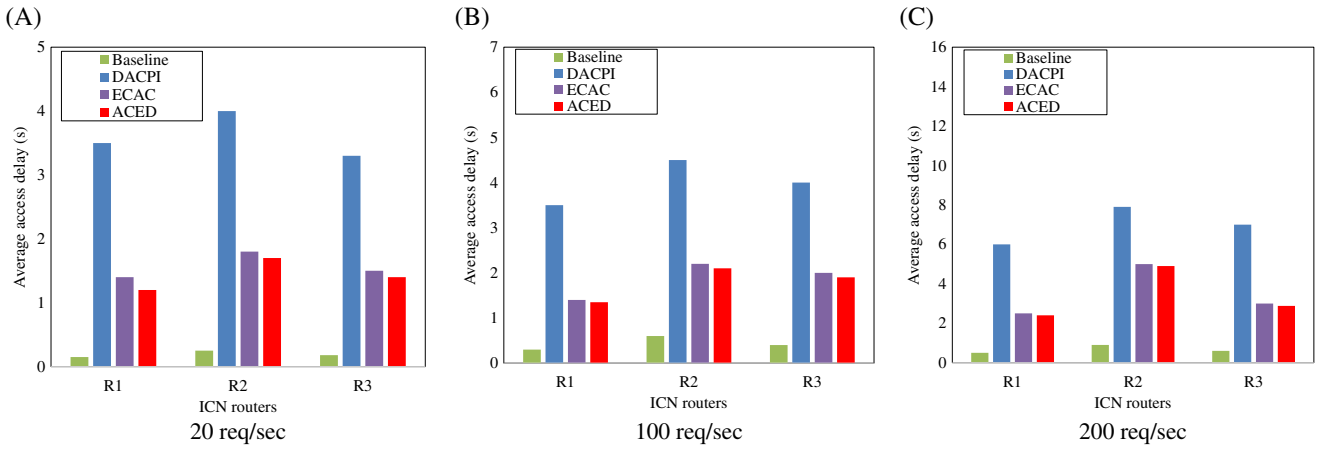

FIGURE 5 ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols under different request rates with cache size = 5000 entry. (a) 20 req/s, (b) 100 req/s and (c) 200 req/s

required overhead. Experimental results show that ECAC achieves results similar to the ACED in all cases. In the experiments, we evaluate the average access time delay caused by our proposed protocol and compare it with related protocols under different scenarios. We study the impact of our proposed protocol on ICN by measuring the average request access time delay, which represent delay between first request sent and data packet received. We evaluate ECAC using ndnSIM, which is a simulator for the NDN architecture within NS-3.[49] NDN architecture uses hierarchal naming scheme. We use the simulator for checking ICN performance with and without the access control schemes.

We build our experiments using backbone AT&T network,[50] which is Internet-like architecture. In our experiments, the network consists of 150 subscribers, 10 publishers, and more than 40 routers as shown in Figure 4. Each edge router is connected to 10 ICN users. We change subscriber's request rates between 20, 100, and 200 request per second and cache size between 1000 and 5000 entries. We also use the following parameters: number of pending interest table entries = 1000 entries, payload size = 1 KB. We use default parameters for point to point links and channels as follows: point-to-point channel data rate = 1 Mbps, point-to-point channel delay = 10 ms. We set key size for DACPI = 1024 bits, Key size for ECAC = 160 bits, Key size for ACED = 80 bits. These key sizes achieve the same security level. As a caching replacement strategy, we use the least frequently used technique to evict the least popular content.

We compare the average request access time delay between a baseline case, DACPI, ECAC, and ACED. In the baseline, we measure ICN request access time delay when there is no add-on solution. DACPI is chosen as a representative of DACPIs for ICN. ACED is selected as a representative for centralized access control mechanisms for ICN. In Figures 5 and 6, results are obtained for different request rates (20, 100, and 200 requests/s) and for different cache sizes (5000 and 1000 entry). In Figure 7, results are obtained for different ratios of attackers to legitimate users (20%, 50%, and 80%) with the same request rates for users and attackers, which is 100 requests/s. The results are recorded at three random edge routers as examples and similar results can be obtained from other routers.

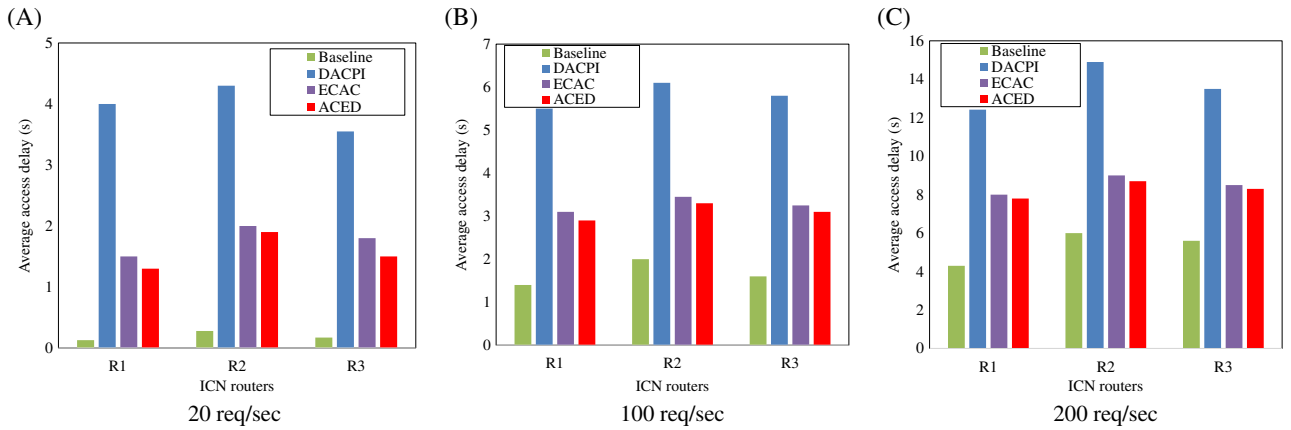

FIGURE 6 ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols under different request rates with cache size = 1000 entry. (a) 20 req/s, (b) 100 req/s and (c) 200 req/s

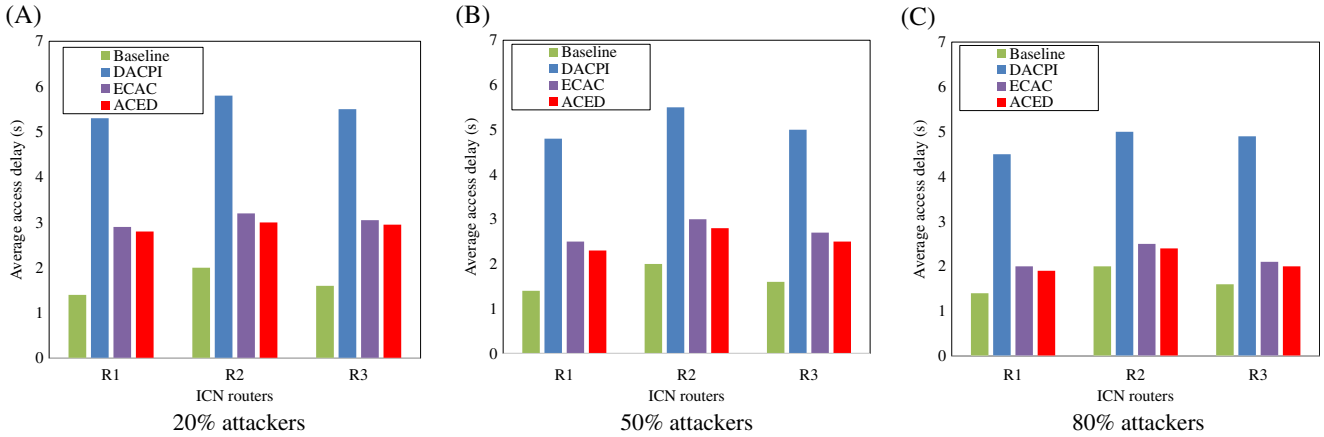

FIGURE 7 ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols under different ratios for attackers to legitimate users with cache size = 1000 entry and request rate = 100 req/s. (a) 20% attackers, (b) 50% attackers, and (c) 80% attackers

3.2 | Results

The experimental results show that the three access control mechanisms increase request access time delay in all cases because of the extra time needed for enforcing access control in ICN. This extra time is coming from either the exchange of access control messages between different entities as in ACED or applying the cryptographic techniques as in ECAC and DACPI. ECAC and ACED achieve similar results in all cases and outperform DACPI. The reason behind almost similar results between ECAC and ACED comes from the difference between the number of messages required for access control. Although in ACED, there is no encryption and decryption as in ECAC, but the extra messages and exchange of these messages between the extra entities in ACED minimize the time difference between the two protocols. Based on Figures 5 and 6, as the cache size increases, request access time delay for all cases and the difference between the three access control protocols and baseline decrease. It happens because the number of cache content evictions decreases as the cache size increases, and hence routers can use their local copies instead of sending requests to the original sources. As depicted in Figure 5a, the average request access time delay is recorded as 0.2, 3.6, 1.5, and 1.4 for the baseline, DACPI, ECAC, and ACED, respectively. When 100 requests per second with cache size of 5000 entry occur, as shown in Figure 5b, the values of average access delay become 0.4, 4, 1.8, and 1.7 for the same protocols. The average access time delay is 0.6 seconds for the baseline, 6.9 seconds for DACPI, 3.5 seconds for ECAC, and 3.3 seconds for ACED, as in the last subcase as depicted in Figure 5c.

As the number of requests increases, the required access time delay gradually increases. In Figure 6a, the average access time delay is 0.2, 4, 1.7, and 1.5 seconds for the baseline, DACPI, ECAC, and ACED, respectively. The average access time delay is 1.6 seconds for the baseline, 5.8 seconds for DACPI, 3.2 seconds for ECAC, and 3.1 seconds for ACED, as depicted in Figure 6b. Figure 6c shows that the average access time delay is 5.3, 13.6, 8.5, and 8.2 seconds for the same protocols.

In the normal case, when there is no attack as shown in Figure 6, the differences between the baseline and the three access control mechanisms are more than the differences in the attack case. Also, as the number of attackers increases, the three protocols achieve better results with respect to the baseline, as shown in Figure 7. In the attack case, the three protocols succeed

to detect attackers and reduce access control messages and related computations. Attackers send uniformly distributed malicious requests, which means the same number of requests for each content. When 20% attackers exist, as shown in Figure 7a, the average access time delay is 1.6, 5.5, 3, and 2.9 seconds for baseline, DACPI, ECAC, and ACED, respectively. When 50% attackers exist as depicted in Figure 7.b, the average access time delay is 1.6, 5.1, 2.7, and 2.5 seconds for the same protocols. The average access time delay is 1.6 seconds for the baseline, 4.8 seconds for DACPI, 2.2 seconds for ECAC, and 2.1 seconds for ACED, as shown in Figure 7c when 80% attackers exist.

4 | COMPARISON WITH EXISTING ICN ACCESS CONTROL PROTOCOLS

In addition to the related work discussed in Section 2, we make a direct comparison between ECAC and the related centralized and decentralized access control protocols. As shown in Table 2, we use the following criteria as a basis for comparison: number of public authentication messages, number of extra entities, number of secret keys, number of ICN names for each content, availability of content names, required security techniques, authentication type, access control decision, and ICN architecture.

Table 2 indicates that DACPI and ECAC have the minimum number of public messages and do not require any additional modifications to the architecture. In Reference 1 eight public messages are needed to enforce the access control for both publication and subscription functions. In Reference 2 a total of eight messages are needed for access control distributed as four messages for publication and four for subscription. In Reference 1 two extra entities are needed, while in Reference 2 one extra entity is needed for access control purposes. In SAC,[3] a total of 11 messages are needed for publication and subscription. SAC does not need extra entities to ICN architectures.

For the required secret information, DACPI and ECAC use the same or fewer secret keys than the other protocols. In both protocols, each user has a public and a private key. ECAC uses smaller key size to achieve the same security level of DACPI. All the four access control mechanisms use only one name for each ICN content except for the decentralized mechanism[3] that uses two names for each content. The content names are available to all users in all protocols, except for SAC,[3] the name is known to legitimate users only.

DACPI uses extra cryptographic techniques with respect to the other protocols, while ECAC uses less cryptographic techniques than DACPI and similar to the decentralized mechanism.[3] Although the number of parameters in ECAC is more than DAPCI, the required space in ECAC is less because ECC can achieve similar security level with smaller key sizes relative to RSA. We use key size of 160 bits in ECAC to have the same security level of 1024 bits key size in DACPI. ECAC provides faster computations because messages generated from ECAC are shorter and require less space than DACPI. These advantages enable ECAC to decrease these extra overhead for ICN architectures.

Fotiou et al.[1] design their solution for PURSUIT ICN architecture, while Aiash et al.[2] propose their solution for network of information (NetInf) ICN architecture. The other three decentralized mechanisms can be applied to different ICN architectures. Centralized access control mechanisms need to be bounded to a specific architecture because they need architectural modification by adding extra entities, while decentralized ones can be adapted to be implemented in different architectures. Apparently, this is a major advantage of decentralized schemes over the centralized ones in ICN architectures.

In terms of the impact on ICN performance, ECAC achieves results similar to centralized access control scheme representative (ACED) without the need of adding new entities or architectural modification to ICN architecture. Also, ECAC is superior over a representative decentralized access control scheme (DACPI) in all cases.

As mentioned in the related work section, many works use encryption-based techniques for achieving access control. Encryption-based access control techniques require either modifications to contents or ICN requests to encrypt the transmitted data. Additionally, encryption-based techniques require high computational overhead, because they need to encrypt high volumes of contents or requests. Encrypting and decrypting high volume of data and requests themselves may not be suitable for constrained capability devices that will exist in the upcoming Internet of Things. ECAC depends on one of the existing ICN naming schemes instead of proposing a new or modified version of ICN naming or requests. In ECAC, we do not encrypt contents or subscription requests, we exchange them as proposed by ICN architectures. We only encrypt and secure the parameters required for access control purposes.

5 | CONCLUSION

The expected number of users, devices, applications, and contents in the Next Generation Internet (NGI) will exceed the capabilities of the current Internet architecture. ICN is one of the proposed architectures for the future Internet that changes the Internet from host oriented to content oriented. New types of attacks have appeared in ICN, in addition to legacy attacks that have greater impact on this new paradigm. In ICN, access control enforcement becomes a major problem because contents are highly distributed and cached everywhere in ICN architectures and subscribers can access these contents from any available

TABLE 2 Comparison between ECAC and existing centralized and decentralized access control protocols

	Access control enforcement delegation (ACED)[1]	Authentication and authorization approach[2]	Session-based authentication (SAC)[3]	Decentralized access control protocol for ICN (DACPI)[4]	Elliptic curve based access control for ICN (ECAC)
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	Five for both subscription and publication
Number of extra entities	Two (AC provider and relaying party)	One (trusted ticket granting)	Zero	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	One secret key and each user has public and private keys. Other secrets do not need to be exchanged.
Number of ICN names for each content	One	One	Two (public and secure name)	One	One
Availability of content names	Available to all users	Available to all users	Available to legitimate users	Available to all users	Available to all users
Required security techniques	No extra techniques are required	Public key infrastructure and hashing	Public key infrastructure, hashing, and random number generation	RSA, hashing, Diffie-Hellman, and random number generation	Elliptic curve, hashing, and random number generation
Authentication type	One-way (access control provider authenticates ICN)	One-way (name resolution service authenticates ICN)	Mutual authentication between ICN nodes and ICN users	Mutual authentication between ICN nodes and ICN users	Mutual authentication between ICN nodes and ICN users
Access control	Centralized	Centralized	Decentralized	Decentralized	Decentralized
ICN architecture	Publish Subscribe Internet Technology (PURSUIT)	Network of Information (NetInf)	Can be applied to different ICN architectures	Can be applied to different ICN architectures	Can be applied to different ICN architectures

copy. ICN requires an access control solution that should be integrated with the architecture itself not as an overlay security layer as the existing solutions for the Internet.

In this paper, we propose an EC-based Access Control protocol for ICN (ECAC) to achieve access control security service to ICN contents. The security of ECC depends on the difficulty of solving the EC logarithm problem. ECAC depends on EC, hashing, and random number generations. ECAC protocol is a decentralized access control mechanism, which does not require new entities to ICN architectures. Access control decisions in ECAC are taken by ICN nodes and subscribers themselves without central authority. Using ECAC, only authenticated users can access authenticated contents.

The proposed protocol successfully prevents man-in-the-middle, 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 protocols. The attacker cannot form a valid authentication messages without knowing the secret keys. ECAC also efficiently minimizes the number of public messages and storage overhead for the access control process. With the light communication, computational, and storage overhead used in ECAC, the access control security level is remarkably increased. Based on our experiments, ECAC uses fewer public messages and does not require extra entities as in the centralized mechanisms. We measure the average request access time delay that indicates the roundtrip time between a first request and a data packet. The importance of this metric is due to the one-to-one relation between requests and data packets in ICN. Our results show that ECAC achieves better performance results for ICN architectures than competitive decentralized mechanisms and achieve similar results with the centralized ones. To achieve access control security service in ICN, we highly recommend using decentralized access control based on ECC. This option can achieve the best performance results in the existence of malicious unauthorized access actions without architectural modifications.

The future work stemming from this paper is to prevent another type of attacks in ICN, which is related to privacy issues in ICN architectures. An attacker can attract user requests to know private information about content popularities and who requested these contents. An attacker can also monitor certain ICN content names to get information about content popularities and block-specific contents. Another direction is to develop a comprehensive security framework for ICN architectures to address different types of naming, routing, caching, and miscellaneous attacks. This framework will contain the required functions in ICN subscribers, publishers, and routers.

ORCID

Eslam G. AbdAllah http://orcid.org/0000-0001-7669-3016

REFERENCES

1. Fotiou N, Marias GF, Polyzos GC. Access control enforcement delegation for information-centric networking architectures. Paper presented at: The Second Edition of the ICN Workshop on Information-Centric Networking, ACM; 2012; 85–90.
2. Aiash M, Loo J. An integrated authentication and authorization approach for the network of information architecture. *J Netw Comput Appl*. 2014;50:73-79.
3. Wang Y, Xu M, Feng Z, Li Q, Li Q. Session-based access control in information-centric networks: design and analyses. Paper presented at: IEEE Performance Computing and Communications Conf (IPCCC); December 2014; Austin, TX; 1–8.
4. AbdAllah EG, Zulkernine M, Hassanein HS. DACPI: A decentralized access control protocol for information centric networking. Paper presented at: IEEE Symposium on Communication and Information System Security (ICC '16); May 2016; Kuala Lumpur, Malaysia.
5. Kurihara J, Uzun E, Wood CA. An encryption-based access control framework for content-centric networking. IFIP Networking. 2015. https://www.ietf.org/mail-archive/web/icnrg/current/pdf9_JIQ5GBeS.pdf. Accessed June 6, 2018.
6. Ghali C, Schlosberg MA, Tsudik G, Wood CA. Interest-based access control for content centric networks (extended version). Paper presented at: ACM Conference on Information-Centric Networking (ICN'15); 2015; San Francisco, CA; 1–10.
7. Misra S, Tourani R, Majd NE. Secure content delivery in information centric networks: design, implementation, and analyses. Paper presented at: ACM SIGCOMM Workshop on Information-Centric Networking (ICN'13); 2013; Hong Kong, China; 73–78.
8. Ion M, Zhang J, Schooler EM. Toward content-centric privacy in ICN: attribute-based encryption and routing. Paper presented at: ACM Symposium on Information, Computer and Communications Security (ASIA'CCS13); 2013; Hangzhou, China; 513–514.
9. Wood CA, Uzun E. Flexible end-to-end content security in CCN. Paper presented at: IEEE 11th Conference on Consumer Communications and Networking (IEEE CCNC); January 2014; Las Vegas, NV; 858–865.
10. Smetters DK, Golle P, Thornton JD. CCNx access control specications, PARC, Technical Report, 2010. http://www.arl.wustl.edu/~jdd/NDN/NDN_GEC/ccnx/doc/specs/AccessControl/AccessControlSpecs01.pdf. Accessed June 6, 2018.
11. Li B, Verleker A P, Huang D, Wang Z, Zhu Y. Attribute-based access control for ICN naming scheme. Paper presented at: IEEE Conference on Communications and Network Security (CNS); 2014; San Francisco, CA; 391–399.
12. Rembarz R, Catrein D, Sachs J. Private domains in networks of information. Paper presented at: IEEE International Conference on Communications Workshops; June 2009; Dresden, Germany; 1–5.
13. Guo H, Wang X, Chang K, Tian Y. Exploiting path diversity for thwarting pollution attacks in named data networking. *IEEE Trans Inf Forensic Secur*. 2016;11(9):2077-2090.
14. Vahlenkamp M, Whlisch M, Schmidt TC. Backscatter from the data plane - threats to stability and security in information-centric networking. *Comput Netw*. 2013;57(16):3192-3206.
15. AbdAllah EG, Zulkernine M, Hassanein HS. Countermeasures for mitigating ICN routing related DDoS attacks. Paper presented at: The 10th International Confernce on Security and Privacy in Communication Networks (Securecomm14); 2014; Beijing, China; 84–92.
16. AbdAllah EG, Zulkernine M, Hassanein HS. Detection and prevention of malicious requests in ICN routing and caching. Paper presented at: The 13th IEEE International Conference on Dependable, Autonomic, and Secure Computing (DASC-2015); 2015; Liverpool, UK; 1741–1748.

17. Xie M, Widjaja I, Wang H. Enhancing cache robustness for content-centric networking. Paper presented at: Proceedings of the IEEE Infocom; March 2012; Orlando, FL; 2426–2434.
18. Fotiou N, Marias GF, Polyzos GC. Fighting spam in publish/subscribe networks using information ranking. Paper presented at: 6th EURO-NF Conference on Next Generation Internet (NGI); June 2010; Paris, France; 1–6.
19. Ghali C, Tsudik G, Uzun E. Needle in a haystack: mitigating content poisoning in named-data networking. Paper presented at: NDSS Workshop on Security of Emerging Networking Technologies (SENT'14); 2014; San Diago, CA.
20. Gasti P, Tsudik G, Uzun E, Zhang L. DoS & DDoS in nameddata networking. Paper presented at: Proceedings of the 22nd International Conference on Computing Communications and Networks (ICCCN); August 2013; Nassau, Bahamas; 1–7.
21. Afanasyev A, Mahadevan P, Moiseenko I, Uzun E, Zhang L. Interest flooding attack and countermeasures in named data networking. Paper presented at: IFIP Networking Conference; 2013; New York, NY; 1–9.
22. Compagno A, Conti M, Gasti P, Tsudik G. Poseidon: mitigating interest flooding DDoS attacks in named data networking. arXiV:1303.4823v3 [cs.NI], 2013. http://arxiv.org/pdf/1303.4823.pdf. Accessed June 6, 2018.
23. Conti M, Gasti P, Teoli M. A lightweight mechanism for detection of cache pollution attacks in named data networking. *Comput Netw*. 2013;57(16):3178-3191.
24. Stallings W. *Cryptography and Network Security: Principles and Practice*. 6th ed. USA: Prentice Hall; 2013.
25. Wong W, Nikander P. Secure naming in information-centric networks. Paper presented at: ACM Workshop of the Re-Architecting the Internet (ReArch'10); 2010; Philadelphia, PA.
26. Zhang X, Chang K, Xiong H, Wen Y, Shi G, Wang G. Towards name-based trust and security for content-centric network. Paper presented at: 19th IEEE International Conference on Network Potocols; 2011; Vancouver, BC; 1–6.
27. Mohaisen A, Mekky H, Zhang X, Xie H, Kim Y. Timing attacks on access privacy in information centric networks and countermeasures. *IEEE Trans Depend Secure Comput*. 2015;12(6):675-687.
28. Burke J, Horn A, Marianantoni A. Authenticated lighting control using named data networking. NDN Technical Report No. NDN-0011, Los Angeles, CA: University of California; 2012. http://named-data.net/wp-content/uploads/TRlighting.pdf. Accessed June 6, 2018.
29. He D, Zeadally S. An analysis of RFID authentication schemes for Internet of things in healthcare environment using elliptic curve cryptography. *IEEE Internet Things J*. 2015;2(1):72-83.
30. Lee YK, Sakiyama K, Batina L, Verbauwhede I. Elliptic-curve-based security processor for RFID. *IEEE Trans Comput*. 2008;57(11):1514-1527.
31. Liu Z, Seo H, Großschädl J, Kim H. Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes. *IEEE Trans Inf Forensic Secur*. 2016;11(7):1385-1397.
32. Du X, Guizani M, Xiao Y, Chen HH. A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Trans Wireless Commun*. 2009;8(3):1223-1229.
33. Azarderakhsh R, Jarvinen KU, Mozaffari-Kermani M. Efficient algorithm and architecture for elliptic curve cryptography for extremely constrained secure applications. *IEEE Trans Circuits Syst*. 2014;61(4):1144-1155.
34. Tawalbeh L, Mowafi M, Aljoby W. Use of elliptic curve cryptography for multimedia encryption. *IET Inf Secur*. 2013;7(2):67-74.
35. Tseng CH, Wang S, Tsaur W. Hierarchical and dynamic elliptic curve cryptosystem based self-certified public key scheme for medical data protection. *IEEE Trans Reliab*. 2015;64(3):1078-1085.
36. Chabanne H, Tibouchi M. Securing e-passports with elliptic curves. *IEEE Secur Priv Mag*. 2011;9(2):75-78.
37. Maletsky K. RSA vs ECC comparison for embedded systems. Atmel. 2015. http://www.atmel.com/images/atmel-8951-cryptoauth-rsa-ecc-compar∖∖ison-embedded-systems-whitepaper.pdf. Accessed June 6, 2018.
38. Cisco. Cisco visual networking index: forecast and methodology 2015-2020. 2016. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf. Accessed June 6, 2018.
39. AbdAllah EG, Hassanein HS, Zulkernine M. A survey of security attacks in information-centric networking. *IEEE Commun Surv Tutor*. 2015;17(3):1441-1454.
40. Tsilopoulos C, Vasilakos X, Katsaros K, et al. A survey of information-centric networking research. *IEEE Commun Surv Tutor*. 2014;16(2):1024-1049.
41. Ahlgren B, Dannewitz C, Imbrenda C, Kutscher D, Ohlman B. A survey of information-centric networking. *IEEE Commun Mag*. 2012;50(7):26-36.
42. Pan J, Paul S, Jain R. A survey of the research on future internet architectures. *IEEE Commun Mag*. 2011;49(7):26-36.
43. Eriksson A, Ohlman B, Persson K. What are the services of an information-centric network, and who provides them?. Paper presented at: The Fourth International Conference on Advances in P2P Systems (AP2PS 2012); September 2012; Barcelona, Spain; 11–17.
44. Baccelli E, Mehlis C, Hahm O, Schmidt TC, Wählisch M. Information centric networking in the IoT: experiments with NDN in the wild. Paper presented at: ACM Conference on Information-Centric Networking (ICN'14); September 2014; Paris, France; 77–86.
45. Moiseenko I, Wang L, Zhang L. Consumer/producer communication with application level framing in named data networking. Paper presented at: ACM Conference on Information-Centric Networking (ICN'15); 2015; San Francisco, CA; 99–108.
46. Bari Md F, Chowdhury SR, Ahmed R, Boutaba R, Mathieu B. A survey of naming and routing in information-centric networks. *IEEE Commun Mag*. 2012;49(12):44-53.
47. Almeida F, Lourenço J. Information centric networks-design issues, principles and approaches. *Int J Latest Trends Comput*. 2012;3(3):58-66.
48. Ghodsi A, Shenker S, Koponen T, Singla A, Raghavan B, Wilcox J. Information-centric networking: seeing the forest for the trees. Paper presented at: Proceedings of the 10th ACM Workshop on Hot Topics in Networks; 2011; 1—6.
49. Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3. NDN Technical Report No. NDN-0005; 2012.
50. Heckmann O, Piringer M, Schmitt J, Steinmetz R. On realistic network topologies for simulation. Paper presented at:ACM Sigcomm; 2003; Karlsruhe, Germany; 28–32.

How to cite this article: AbdAllah EG, Zulkernine M, Hassanein HS. Preventing unauthorized access in information centric networking. *Security and Privacy* 2018;1:e33. https://doi.org/10.1002/spy2.33