Secure Distribution of Protected Content in Information-Centric Networking

Muhammad Bilal, *Member, IEEE*, and Sangheon Pack, *Senior Member, IEEE*

***Abstract*—The benefits of the ubiquitous caching in information centric networking (ICN) are profound; even though such features make ICN promising for content distribution, but it also introduces a challenge to content protection against the unauthorized access. The protection of a content against unauthorized access requires consumer authentication and involves the conventional end-to-end encryption. However, in ICN, such end-to-end encryption makes the content caching ineffective since encrypted contents stored in a cache are useless for any consumers except those who know the encryption key. For effective caching of encrypted contents in ICN, we propose a secure distribution of protected content (SDPC) scheme, which ensures that only authenticated consumers can access the content. SDPC is lightweight and allows consumers to verify the originality of the published content by using a symmetric key encryption. SDPC also provides protection against privacy leakage. The security of SDPC was proved with the Burrows–Abadi–Needham (BAN) logic and Scyther tool verification, and simulation results show that SDPC can reduce the content download delay.**

***Index Terms*—Access control, authentication, content distribution, 5G, information-centric networking, in-network caching, named data networking, privacy, security.**

I. Introduction

SINCE the earliest time of the Internet, its underlying architecture has been based on packet-switching and host-to-host communications. The Transmission Control Protocol/Internet Protocol-layered architecture employs the same view and provides an abstract host-to-host communication model to communication applications. It decouples what to communicate from how the communication is done. This basic design feature of the TCP/IP architecture was far-reaching, allowing the Internet to grow for almost four decades while adopting various features and yet maintaining high efficiency. However, in the recent past, there has been a profound increase in Internet connectivity, and with the emergence of new Internet applications, the Internet semantics have changed from host-centric to content-centric. To satisfy the needs of emerging Internet applications, the current TCP/IP Internet architecture has adopted several application layer solutions known as over-the-top (OTT) applications, such as content delivery network (CDN), web caching, and peer-to-peer networking [1], [2]. With the addition of numerous applications, the gap between the basic semantics of the current Internet architecture and its usage is bound to increase; in fact, the addition of new OTT applications is leading us toward a very complex Internet architecture, and are introducing challenges to achieving efficiency, security, and privacy at acceptable economical cost.

Further, Internet trends are shifting away from browsing information to online consuming and sharing all types of content, including user-generated contents. Hence, the most promising characteristic of the future Internet is ubiquitous content delivery. What is being communicated is becoming more important than who is communicating. In this perspective, information-centric networking (ICN) has emerged as a promising architecture for the Future Internet; recently, the ICN support for 5G use cases were specified by next generation mobile networks (NGMN). ICN represents a paradigm shift from host-centric to content-centric services and from source-driven to receiver-driven approaches. In the ICN paradigm, the network is in charge of doing the mapping between the requested content and where it can be found. To do so, a network-level naming is used for identifying content objects, independent of their locations [3]–[5]. This means that the ICN architecture decouples contents from the host at the network level and supports a temporary storage of contents at in-network caches.

In ICN, in-network caching is an integral part of the ICN service framework [6], [7]. The benefits of the ubiquitous caching in ICN are profound, but it also introduces a challenge to content security; especially, the protection of a private or confidential content is a challenging task. The ICN-enabled cache routers can store the content segments for future use; hence, the content is temporarily cached in few intermediate cache routers while it is being delivered to a consumer. If content requests traverse a cache router that holds a temporarily cached copy of that particular content segment, then the request is entertained locally without being routed toward the publisher. However, in ICN, the publisher has no control over the content after injecting it in the network; in particular, if a private or confidential content is protected insecurely, then any unauthorized consumer can acquire it from intermediate caches. Traditionally, the protection of the content against unauthorized access requires consumer authentication and involves the conventional end-to-end encryption. Consequently, when the content is encrypted with the

Manuscript received January 4, 2019; revised May 26, 2019; accepted July 25, 2019. Date of publication August 14, 2019; date of current version June 3, 2020. This work was supported in part by the Hankuk University of Foreign Studies research funds for 2019 and in part by National Research Foundation (NRF) of Korea Grant funded by the Korean Government (MSIP) under Grant 2017R1E1A1A01073742. This paper was presented in part at the IEEE VTC 2018-Spring RAFNET Workshop, Porto, Portugal, June 2018 [41]. *(Corresponding author: Sangheon Pack.)*

M. Bilal is with the Hankuk University of Foreign Studies, Department of Computer and Electronics Systems Engineering, Yongin-si, Gyeonggi-do 17035, South Korea (e-mail: m.bilal@ieee.org).

S. Pack is with School of Electrical Engineering, Korea University, Anam-dong, Seoul 02841, South Korea (e-mail: shpack@korea.ac.kr).

Digital Object Identifier 10.1109/JSYST.2019.2931813

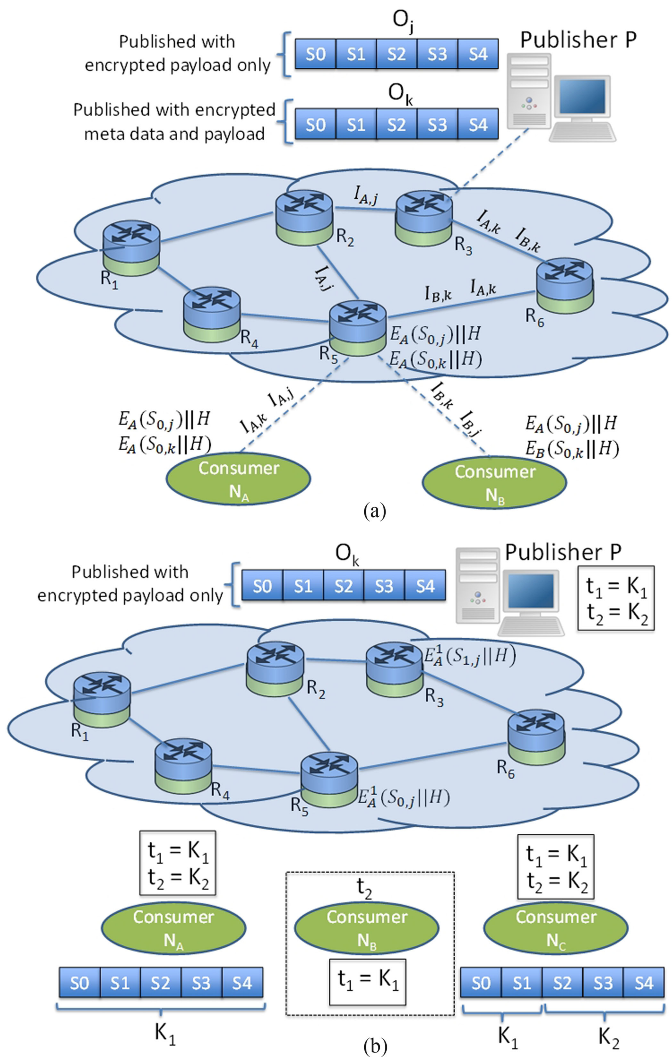


Fig. 1. Ineffective caching in ICN with end-to-end encryption.

authorized user's key, the in-network caching becomes ineffective in ICN.

In Fig. 1, a publisher P publishes two content objects O_j and O_k. Further, two consumers N_A and N_B subscribe to access these protected contents O_j and O_k. Furthermore, the object O_j is published without encrypting, scrambling, or hashing the content name, while the object O_k is published with encrypting, scrambling, or hashing the content name to ensure its privacy. Assume that the consumer N_A sent an interest packet $I_{A,j}$ encapsulating the access authorization information. In reply, based on the subscription information, if N_A is a valid subscriber, the publisher P encrypts the requested content segment $S_{0,j}$ with a consumer-specific key and sends it to the consumer N_A. Based on the semantics of ICN and cache replacement schemes, the intermediate cache router R_5 stores the encrypted content segment $S_{0,j}$ for future use. Now let's suppose, the consumer N_B requests the same content. If the meta-data of the encrypted stored packet is available to R_5, as in case of O_j, then the intermediate cache router R_5 will reply with the cached content $S_{0,j}$ to the consumer N_B. However, N_B cannot decrypt the content segment $S_{0,j}$ as it was solely intended for N_A and thus the payload is encrypted with the key known to P and N_A. Contrarily, if the meta-data of the encrypted stored packet is unavailable to R_5, as in case of O_k, the interest packet $I_{B,k}$ will be forwarded to the publisher.

This issue can be solved by encrypting each content segment with a key known to all subscribers. In this regard, this issue can be viewed as a group key agreement problem. However, even in the presence of a perfect key distribution protocol, the assurance of backward and forward secrecy requires complex operations since the publisher in ICN has no control over the content after injecting it in the network. Moreover, in conventional group key agreement protocols [8], [9], the hosts share a cryptographic key for secure communications, which are not well suited for the content-centric ICN paradigm.

For example, if an authorized consumer unsubscribes from the service, then to ensure the forward secrecy, it is necessary to make sure that leaving consumer does not have access to future keys for the group; hence, the shared key should be updated. From this point onward, the publisher would encrypt new version of content with updated group keys. To access the content which is already disseminated in network caches, the authorized consumers need to keep both keys for effective cache utilization. As shown in Fig. 1(b), at time t_1, the publisher P publishes object O_j and shares the encryption key K_1 with all authorized consumers N_A, N_B, and N_C. Let's assume at time t_2, consumer N_B unsubscribes with publisher P. Publisher P will issue a new key K_2. Further assume that before unsubscribing an event, the copies of segment $S_{0,j}$ and $S_{1,j}$ were already disseminated in ICN core network; now if a consumer N_A or N_C requests object O_j, it may get $S_{0,j}$ and $S_{1,j}$ from cache router encrypted with K_1 and rest of the segments from publishers encrypted with K_2. Similarly, if a new authorized consumer subscribes for the service, then to ensure the backward secrecy, the shared key should be updated, and previous group members need to keep both keys for an effective cache utilization. Imagine a highly dynamic group where the consumers subscribe or unsubscribe very frequently; it will trigger numerous leave and join events, which will invoke group key agreement protocols each time. For effective caching, all consumers would keep record of multiple keys. Moreover, an extra decision operation is required to select a proper key; associating a time stamp can solve the problem at the cost of group member synchronization. Hence, the conventional group key management cannot handle the access control problem in ICN for ensuring the effective caching.

In our proposed scheme, we shifted the central target of keying process from hosts to data itself, i.e., the segments of the published content are encrypted with symmetric cryptographic keys that are unique to each segment and versions. The solution is to encrypt each content segment with a uniquely assigned key known to all subscribers; this raises three fundamental questions. How does one ensure that only an authenticated subscribed consumer can access the content? How can the consumer verify the originality of the content; that is, do we still need self-certifying? Finally, and most importantly, how can encryption keys be distributed among all of the consumers for each content segment? We answered all these questions in this paper.

Specifically, we propose a secure distribution of protected content (SDPC) scheme, which consists of two protocol suites:

1) the keying protocol suite and 2) the subscription and content access protocol suite. The keying protocol suite enables the consumer and publisher to share a chain of secret keys required to decrypt the segments of the published content, while the subscription and content access protocol suite ensures that only authorized consumers receive the secret key generation information.

The remainder of this paper is organized as follows. In Sections II and III, we summarize the related works and present the system model, respectively. Section IV describes SDPC with detailed discussions. Section V presents an inclusive security analysis. In Section VI, we present the performance analysis of SDPC. Finally, we provide concluding remarks in Section VII.

II. Related Works

Most existing access control schemes for secure contents are application specific or lack security strength. For example, in [10], the authors presented a scheme for protected contents using network coding as encryption. However, the scheme requires a private connection between the publisher and consumer to obtain the decoding matrix and missing data blocks. In [11], the authors presented a security framework for the copyrighted video streaming in ICN based on linear random coding. It is proven that the linear random coding alone improves the performance of ICN [12]. However in [11], each video was encrypted with a large number of symmetric encryption keys, such that each video frame was encrypted with a unique symmetric encryption key. Since only authorized users who possessed the set of all keys could decrypt the video content, the distribution of a large number of keys for each video content can be an extra communication overhead.

In earlier work [13], the authors proposed a content access control scheme for ICN-enabled wireless edge. The proposed one is an extension of [14], which employs the public-key-based algorithm and Shamir's secret sharing as a building block, named AccConF. To obtain a unique interpolating polynomial of Shamir's scheme, AccConF espoused Lagrangian Interpolation technique. The calculation of Lagrangian Interpolation is a computationally expensive process. To reduce the client-side computational burden, the publisher piggy backs an enabling block with each content, which encapsulates partially solved Lagrangian coefficients.

In a work [15], an access control is realized by a flexible secure content distribution architecture, which combines the proxy re-encryption and identity-based encryption mechanisms. The publisher generates a symmetric key and encrypts the content before dissemination. To access the content from in-network cache or directly from publisher, a consumer first sends a request to publisher to acquire the symmetric encryption key. Upon receiving the key request, the publisher validates and verifies the authenticity of consumer, and sends the symmetric key encapsulated in response message encrypted with consumers identity. The proposed scheme eliminated the asymmetric encryption, but it is not clear that how the consumer's private identity could be known to the content provider.

In another work [16], the author proposed a content access control scheme based on proxy re-encryption. In proxy re-encryption, the content is re-encrypted by an intermediate node. In the proposed scheme, the edge routers perform the content re-encryption. Upon receiving a content request, the publisher encrypts the data and a randomly generated key k1, using its public key. Upon receiving the content request, edge router generates a random key k2 encrypted by the publishers public key and signed by the edge router. Edge router sends the encrypted k2 to publisher and appends the encrypted k2 with the content and dispatch it toward consumer. Meanwhile, the publisher verifies the authenticity of consumer, and generates the content decryption key K using K1, K2, and public key. Upon receiving K, the consumer can decrypt the content.

In other work [17], the authors proposed a distributed information flow control mechanism (named MCAC) to enable secure access control for the published content. In MCAC, the requests and content objects are labeled with $\{h, n, d, p\}$. These labels classify the contents based on the security and privacy requirements, where the h-level signifies the highest protection level and enforces noncaching policy, the n-level enforces the 1-level caching policy, the d-level permits multilevel caching policy, and the p-level supports all reading policy. To administer the MCAC information flow, the intermediate routers require to implement a trust computing base (TCB), consisting of three modules: trust storage module (TSM), trust labeling module (TLM), and trust enforcement module (TEM). TSM governs the process of cryptographic session key establishment between participating routers and other nodes. The session keys are used to attain the h-level security by encrypting highly confidential h-labeled contents. TLM checks the label value and instructs the operating system accordingly to take further actions. TEM performs the content-forwarding process and is responsible for content reclassification, i.e., TEM can relabel a content to the h-level if it was at the n-level to hide the content based on privacy policy of the publisher. MCAC does not provide any mechanism to authenticate participating entities, which makes MCAC vulnerable to various attack. Moreover, to enforce the h-level security and privacy protection, all MCAC-enabled routers need to establish a cryptographic session key and need to encrypt/decrypt all the communication between routers, which severely affects the performance of MCAC.[1]

In another study [18], the authors presented an access control scheme for the encrypted content in ICN, which is based on the efficient unidirectional proxy re-encryption (EU-PRE) proposed by [19]. The proposed scheme, named efficient unidirectional re-encryption (EU-RE), simplifies EU-PRE by eliminating the need of proxies in the re-encryption operation. However, the EU-RE scheme is still based on asymmetric cryptography, which is not suitable for several resource-constraint applications such as IoT and sensor networks. Moreover, the authors made an assumption that the content provider behaves correctly, i.e., it does not distribute any private content or decryption rights

[1]To verify the protocol claims, we implemented MCAC in an automated security protocol analysis tool, Scyther [21], and also discussed its performance in Sections V and VI.

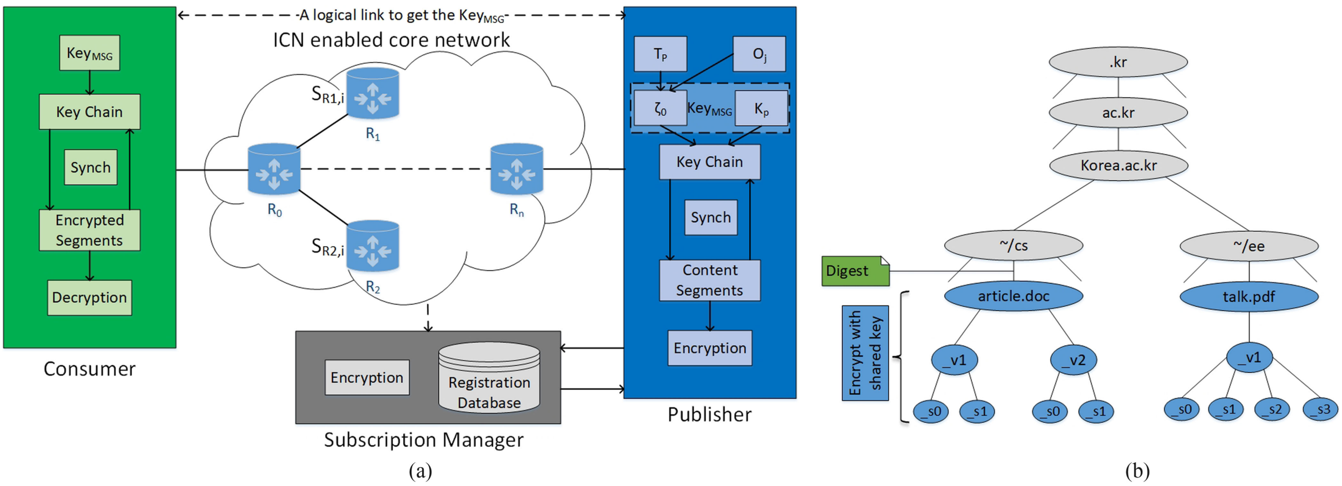


Fig. 2. Illustration of (a) system model and (b) naming scheme used in SDPC.

to unauthorized users. However, this assumption falsifies the protocol claims defined in [20], which means EU-RE is weak against several attacks.[2]

III. System Model

The system model used throughout this work is shown in Fig. 2(a). For concrete discussion and better understanding, in the rest of the paper, we present SDPC for a particular ICN architecture, i.e., named data networking (NDN) [3]. However, SDPC can be adopted for other ICN architectures without changing the core idea.

In NDN core networks, we introduce a new entity, designated subscription manager M. The subscription manager M can be a module installed on the publisher or it could be an independent entity in the network. In this work, we assume that subscription manager M is an independent entity associated with multiple publishers. We also assume that there is a secret number n_S^i associated with each valid subscriber (or consumer) i, which is registered with the subscription manager M. The registration could be made offline or online using a smart mechanism. The subscription manger M stores the secret number n_S^i in a hash table, which is a part of registration database, as shown in Fig. 2(a). Note that being registered does not mean the consumer is entitled to access a certain protected content. When a registered consumer is interested in a protected content, the consumer should first subscribe to the protected content, for instance, subscribing to a movie channel. In the first step, the consumer sends an interest request for the protected content along with the subscription request, and the publisher routes the request toward the subscription manager M. After that, the subscription manager M authenticates the consumer N_A and, in response, the publisher P sends the encryption key generation information $\mathrm{KEY}_{\mathrm{MSG}}$. Using $\mathrm{KEY}_{\mathrm{MSG}}$ as a seed for a secure hash function, the consumer N_A and the publisher P can generate a chain of keys. Then, the publisher P uses these keys to encrypt the segments of the published content; likewise, after acquiring $\mathrm{KEY}_{\mathrm{MSG}}$, the consumer N_A generates the same keys to decrypt the segments of the published content.

To acquire $\mathrm{KEY}_{\mathrm{MSG}}$, the first interest packet sent by a consumer should reach the publisher. To avoid any cache hit, it is important the name of the content should be unique between the consumer and the publisher, yet it should identify the requested object. As shown in Fig. 2(b), the name of the segment 0, "korea.ac.kr/∼fil/test.doc/_v1/_s0," is a variable length and in a human readable form. However, to achieve the name uniqueness, the consumer inserts the hash of the secret number n_S^i, and encrypts the content name with K_{TS}^i. Then, the name of the segment 0 becomes "korea.ac.kr/∼fil/Hash $\left(n_S^i\right)/E_{TS}^i$ (test.doc/_$v1$/_$s0$)." In this naming scheme, the insertion of digest and encryption of naming part provide a consumer–publisher specific unique name and, as a result, the interest packet always reaches the publisher without any cache hit. Note that the usage of consumer name space is restricted for acquiring $\mathrm{KEY}_{\mathrm{MSG}}$ only, this provides prevention against DoS attacks.

After acquiring $\mathrm{KEY}_{\mathrm{MSG}}$, the consumer can access the rest of the segments by using a shared authoritative name space. The name for each segment includes a hash digest $\mathrm{Hash}\left(\mathrm{KEY}_{\mathrm{MSG}}\right)$, and the object name is encrypted with a uniquely assigned key K_l^j, which is generated using $\mathrm{KEY}_{\mathrm{MSG}}$ for each segment l of an object O_j. For example, the name for segment _$s1$ of object O_j is given by "korea.ac.kr/∼fil/Hash $\left(\mathrm{KEY}_{\mathrm{MSG}}\right)/E_1^j$ (test.doc/_$v1$/_$s1$)." With the insertion of $\mathrm{Hash}\left(\mathrm{KEY}_{\mathrm{MSG}}\right)$ and encryption of the naming part with keys generated using $\mathrm{KEY}_{\mathrm{MSG}}$, this naming scheme provides a shared authoritative name space for all authorized consumers and thus it enables an effective content caching. Moreover, this naming scheme ensures the privacy, because the content name is invisible to outsider without any knowledge of n_S^i, $\mathrm{KEY}_{\mathrm{MSG}}$, and cryptographic keys K_{TS}^i and K_l^j.

Let's suppose Fig. 2(a), a consumer sends an interest packet $I_{A,i}$ utilizing the proposed naming scheme. Then, the packet will reach publisher without any cache hit. Let us say that

[2] To verify the protocol claims, we implemented EU-RE in an automated security protocol analysis tool, Scyther [21], and presented the results in Section V.

protected content object O_i is composed of k segments of $S = \{S_{1,i}, S_{2,i} \dots S_{k,i}\}$; further, the intermediate cache routers R_1 and R_2 have the copies of the protected content segments, represented by $S_{R1,i} \subseteq S$ and $S_{R2,i} \subseteq S$. If the consumer is a valid subscriber, the publisher sends the encryption key generation information KEY_{MSG} to the consumer. After receiving the key generation information, the consumer can decrypt the content segments, which may be delivered directly from the intermediate cache router.

IV. Secure Distribution of Protected Content

SDPC consists of two protocol suites: 1) the keying protocol suite and 2) the subscription and content access protocol suite. The keying protocol suite is composed of a key generation protocol and a key agreement protocol for content protection. Likewise, the subscription and content access protocol further includes three protocols, one dealing with the consumer subscription and the other two dealing with access to the protected contents published by different publishers.

A. Keying Protocol Suite

In the keying protocol suite, the key generation protocol generates a commitment key using an irreversible function similar to the ones used in [22] and [23]. The commitment key is further used to drive multiple keys; for instance, a chain of content segment encryption keys, a ticket encryption key, and a consumer-associated symmetric key are derived from the commitment key.

The key generation mechanism for the content protection is shown in Fig. 3(a). First, the publisher divides a large content into equal sized segments. For each protected content object O_j, the publisher generates a unique commitment key generator by using an irreversible one-way hash function $\zeta_0^j = H(T_P, O_j)$, where T_P is the time of publishing and O_j represents the content name and version.[3] After that, the publisher generates a chain of key generators of the length $L = \frac{\text{sizeof}(O_j)}{\text{segment size}}$ by using an irreversible one-way function $\{H(\zeta_0^j) = \zeta_1^j, H(\zeta_1^j) = \zeta_2^j, \dots, H(\zeta_{l-1}^j) = \zeta_l^j\}$. Each generator ζ in the chain is used by a function g at a specific index location in the chain to derive a content segment encryption key. For instance, at index k, the function $g(\zeta_k^j) = H(\zeta_k^j, K_p)$ generates the key K_k^j used for encrypting the kth segment of the content object O_j, where K_p is the public key of the publisher. The use of K_p, in symmetric key generation process, implicitly ensures the originality of the content, i.e., the content are still self-certifying without use of expensive asymmetric encryption. For instance, very efficient public key algorithms, such as ECC [24], are almost three thousand times slower than symmetric key algorithms [25] such as RC5 [26]. The symmetric keys generated as a result of the SDPC keying protocol have the size of 256 bits. Hence, in the subsequent section on the authentication protocols, any symmetric encryption algorithms supporting the 256-bit key can be used, e.g., RC5/6 [26], Rijndael [27], and Twofish [28].

[3]Each version of content object is encrypted with a separate chain of keys. It empowers the publisher to control version-based access.

B. Subscription and Content Access Protocol Suite

When a consumer wants to subscribe to the protected content, the consumer gains an initial access using a subscription protocol (SubP). After SubP, the consumer can use a ticket to access multiple protected contents published by the publishers or managed by a third party.

1) Initial Access and Subscription Protocol (SubP): If a consumer N_i wants to subscribe to the protected content (e.g., subscribing for a movie channel), N_i first generates an encryption key $K_{TS}^i = H(K_p^j \oplus n_S^i)$, where K_p^j is the public key of the publisher and n_S^i is a secret number shared with the subscription manager M. SubP continues as follows.

1) As shown in **1** in Fig. 3(b), N_i injects a subscription interest packet I_i and the NDN core network forwards it to the publisher P_j. The interest packet encloses n_0^i that is encrypted with the generated encryption key K_{TS}^i.
2) Upon receiving the request from N_i, P_j forwards the request in conjunction with its identity and the challenge n_2 to the subscription manager M. Note that P_j cannot decrypt the part of the interest packet which is encrypted with key K_{TS}^i and registration number n_S^i remains invisible to the publisher.
3) M retrieves the profile of N_i from the database. If N_i is a legitimate consumer, M generates the keys $K_{TS}^i = H(K_p^j \oplus n_S^i)$ and $K_S^i = H(T_M \oplus n_S^i)$, and sends $u_0 = E_{TS}^i\left(n_0 + 1||n_1||T_k||K_s^i\right)$ to P_j, where T_M is the time of issuing the session key K_S^i. The message M3 includes a ticket $T_k = E_P^j\left(N_i||K_s^i||\text{profile}\right)$, a challenge n_1 for N_i, and a challenge response for P_j. After that, P_j verifies the challenge response and stores n_1 to use it as a message authentication in M5 and M6. In addition, P_j retrieves the profile and the session key K_s^i from the ticket. Since ticket is encrypted with the public key of P_j, the consumer N_i cannot decrypt it, but can use it to subscribe to other contents published by P_j, without contacting the subscription manager M.
4) P_j forwards u_0 to N_i along with $\text{KEY}_{\text{MSG}} = H(\zeta_0^j, K_p)$, which is required to decrypt the segments of the published content and also used as a content object identifier. After verification of a challenge $(n_0 + 1)$, N_i accepts T_k and generates a key chain to decrypt the protected content. The generated key chain involves the public key of P_j, and hence, the content is also self-certifying.
5) P_j sends the challenge response $(n_1 + 1)$ to M for the confirmation of a successful protocol run. After challenge $(n_1 + 1)$ confirmation, P_j may optionally register N_i in its own database. If P_j does not receive any challenge response within a certain period of time, P_j marks T_k as a stolen ticket.

In SubP, secure exchanges of n_0, n_1, and n_2 ensure the message authentication between the consumer and the subscription manger, between the subscription manger and the publisher, and between the publisher and subscription manger, respectively. Likewise, the message authentication between the consumer and publisher is established by the session key encryption and n_1.

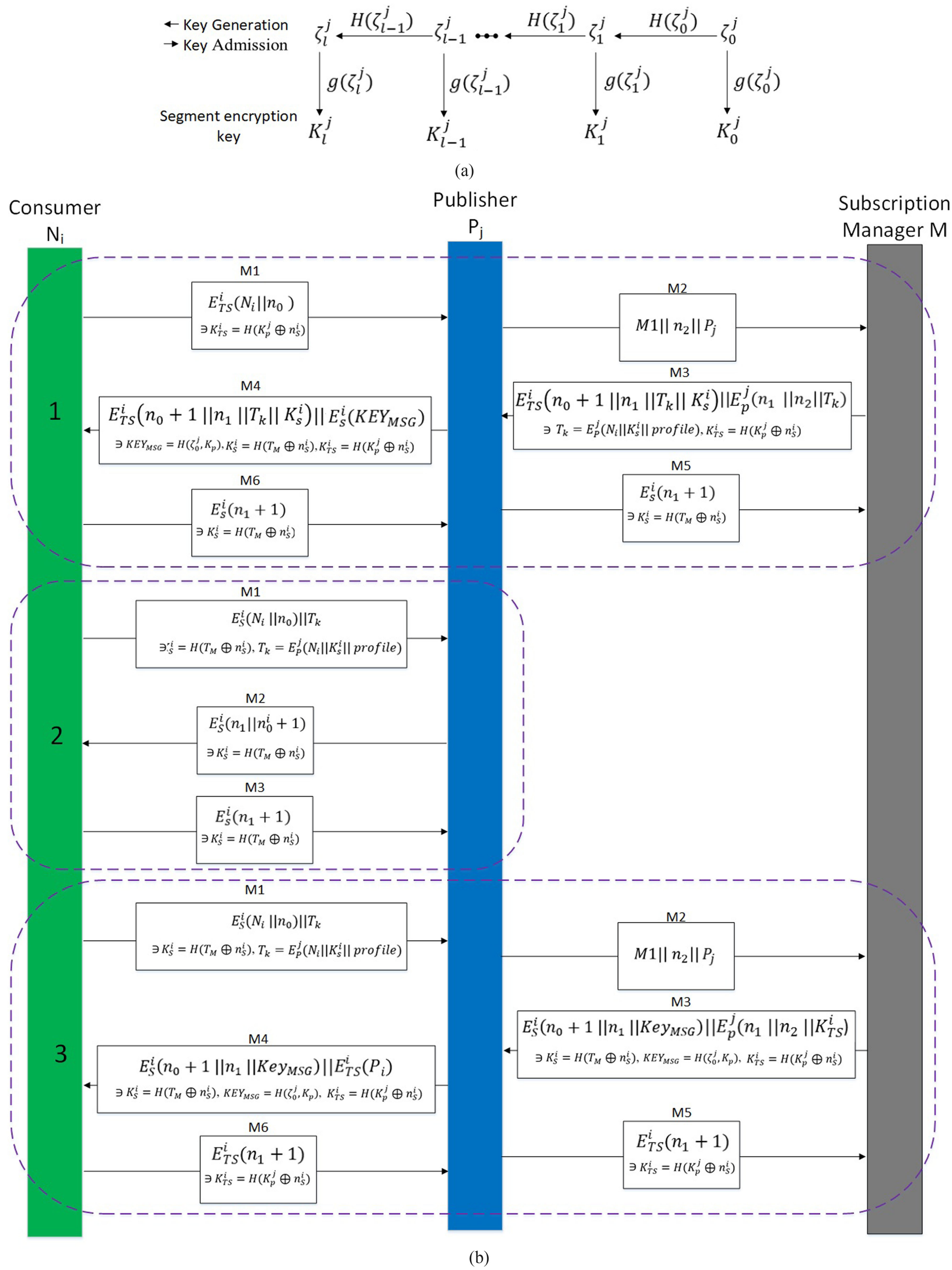


Fig. 3. The SDPC protocol suite. (a) Symmetric keys generation and admission with reference to segment number of protected content. (b) Message exchange for SubP, APSub, and APSub3.

C. Content Access Protocols

1) Access Protocol After Subscription (APSub): When the consumer N_i wishes to access some other protected contents published by the publisher P_j, N_i sends an interest request for the protected content along with the ticket T_k and APSub continues as follows.

1) As shown in **2** at Fig. 3(b), N_i injects a subscription interest packet, enclosing $\text{Access}_{\text{req}} = E_S^i(N_i||n_0)||T_k$. The NDN core network forwards it to the publisher P_j. The publisher P_j decrypts the ticket and verifies the sender's identity N_i by retrieving K_s^i. If the value N_i does not match, P_j will ignore the request and otherwise proceed as follows.
2) P_j sends a challenge response along with the new challenge encrypted with the session key K_s^i. P_j also sends KEY_{MSG}, which is required to decrypt the segments of the published content.
3) N_i sends a challenge response n_1. If P_j does not receive the challenge response within a certain period of time, P_j marks T_k as a stolen ticket.

In APSub, the secure exchange of n_0^i ensures the message authentication between the consumer and the publisher.

2) Access Protocol After Subscription Involving a Third Party (APSub3): Assume a consumer N_i subscribed with P_i, which means it shares a session key K_s^i with P_i and holds a T_k encrypted with public key of P_i. Now if N_i wishes to access the protected contents published by a third-party content publisher P_j, APSub3 continues as follows.

1) As depicted in **3** in Fig. 3(b), N_i injects a subscription interest packet enclosing $\text{Access}_{\text{req}} = E_S^i(N_i||n_0)||T_k$ and the packet is forwarded to the publisher P_j.
2) Upon receiving the request from N_i, P_j forwards the request in conjunction with its identity and the challenge n_2 to M. Note that P_j cannot decrypt $\text{Access}_{\text{req}}$ in the interest packet that is encrypted with the key K_s^i, which is a shared session key between N_i and P_i, which ensures the third-party content distributor cannot misuse the consumer secure information, such as profile and secret share number.
3) M retrieves the profile from T_k, and if N_i is a legitimate consumer, M generates the key $K_{TS}^i = H(K_p^j \oplus n_0)$, and sends $u_0 = E_S^i(n_0+1||n_1||\text{KEY}_{\text{MSG}})$ to P_j. The message M3 also includes the key K_{TS}^i, a challenge n_1 for N_i, and the challenge response for P_j, which are encrypted with the public key P_j. After that, the publisher P_i verifies the challenge response and stores n_1. Note that the ticket is encrypted with the public key of P_i. Therefore, N_i and third-party publisher P_j cannot decrypt it. Also, KEY_{MSG} is inaccessible to P_i, which ensures that the third-party content distributor cannot misuse the protected content.
4) P_j forwards $u_0||E_{TS}^i(P_j)$ to N_i. After the verification of the challenge (n_0+1), N_i generates $K_{TS}^i = H(K_p^j \oplus n_0)$ and sends the challenge response (n_1+1) to P_j. Now N_i can generate a key chain to decrypt the protected published content. Since the key chain is generated using the public key of P_i, the content is also self-certifying.
5) P_j sends the challenge response (n_1+1) to M for the confirmation of a successful protocol run.
6) After the challenge confirmation, P_j closes the protocol run. If P_j does not receive any challenge response within a certain period of time, P_j marks T_k as a stolen ticket.

In SubP3, secure exchanges of n_0, n_1, and n_2 ensure the message authentication between the consumer and the subscription manger, between the subscription manger and the third-party publisher, and between the third-party publisher and subscription manger, respectively. Likewise, the message authentication between the consumer and the third-party publisher is established by a temporary session key K_{TS}^i and n_1.

V. Security Analysis

This section presents an inclusive security analysis, formal analysis using Burrows–Abadi–Needham (BAN) logic [29], and Scyther implementation results [21].

A. Naming-Based Attacks

In NDN, the objects are identified by a human readable naming system, which can lead to watchlist and sniffing attacks [30]–[32].

In watchlist, an attacker, who has control over communication links and cache routers, can delete or filter the content based on a predefined list of content objects. With the use of SDPC, the content is encrypted and invisible to the attacker. Recall that in NDN, it is not obligation that a content object must carry an explicit content name, rather it can carry an implicit content identifier computed from the corresponding interest. This solution hides the object from the attacker. Let us reconsider the example in Fig. 2(b). The first interest packet carries the name "korea.ac.kr/∼fil/ Hash$((n_S^i)$ / $E_{TS}^i(\text{test.doc}/\_v1/\_s0))$"; besides the insertion of hash digest hash$((n_S^i)$, the object name is encrypted with K_{TS}^i . After acquiring KEY_{MSG}, the name for _s1 is then given by "korea.ac.kr/∼fil/ Hash$(\text{KEY}_{\text{MSG}})$ / $E_1^j(\text{test.doc}/\_v1/\_s1)$." The attacker cannot get KEY_{MSG}, n_S^i, K_{TS}^i, and K_l^j, and thus launching watchlist attack is not possible. Moreover, it completely hides the object name from the attacker, which ensures the privacy of the consumer.

Contrarily, in a sniffing attack, the intruder does not have any list of predefined contents, rather it monitors the network and filters or eliminates the data if it contains some specified keywords. Such sniffing attack is not possible in SDPC, because the data are encrypted with the secret keys.

B. DDoS Attacks

The in-network caching makes NDN intrinsically resilient against distributed denial of service (DDoS) attacks [33], [34]. DDoS is a malicious attempt to disturb normal traffic to a server, for instance, multiple compromised systems send fake interest packets to a content publisher. Once the content is disseminated across network caches, the DDoS attack against a publisher depletes due to the on-path cache hits. However, assume somehow an attacker manages to flood all interests to a targeted publisher. With the use of SDPC, the total burden after a successful DDoS attack on the targeted publisher will remain insignificant. This is because the subscription manager M keeps the record of registered nodes in the hash table, in which entries represent $K_{TS}^i = H(K_p^j \oplus n_0)$ session keys. Thus, in case of suspicion, the subscription manger in SDPC can identify fake requests by a hash table lookup with the complexity $O(1)$.

C. Time Analysis Attack

In NDN, any cache node can store content segments. An intruder can guess that a particular content was requested by a user in particular vicinity by observing the request response time of a cached or uncached content. With the use of SDPC, the payload is encrypted with one of the keys derived from $\mathrm{KEY_{MSG}}$ and the name of an object is identified by the digest and encrypted fields. Since the intruder cannot acquire $\mathrm{KEY_{MSG}}$ on time, it cannot create a valid request to launch a time analysis attack.

D. Unauthorized Access

SDPC allows the caches to store encrypted contents and to use a naming scheme unrecognizable to intruders. An intruder can access the content only after acquiring $\mathrm{KEY_{MSG}}$. Since the delivery of $\mathrm{KEY_{MSG}}$ in SDPC is achieved by handshake messages, where each message exchange contains an explicate (nonce challenge) or implicit (encryption key derived from nonce) message authentication, further each message is encrypted with $K_{TS}^i = H(K_p^j \oplus n_0)$; for unauthorized access, an intruder needs to acquire K_{TS}^i.

E. Traffic Monitoring Attack

In traffic monitoring attack [35], an intruder targets a consumer and tries to identity the requested contents. To launch a traffic monitoring attack, the intruder takes control of edge router and observes all the requests send by the target consumer. However, in SDPC, the content name is encrypted, which hides the object name from the attacker, and consequently, the traffic monitoring cannot reveal the identity of requested contents.

F. Formal Analysis Using BAN Logic

BAN logic [29] is widely used for the formal analysis of security protocols, till recently [38], [39]. To verify the security of the SDPC protocol suite, it is sufficient to demonstrate the security of SubP since the rest of the protocols are extensions of SubP. The BAN logic analysis shows that SDPC is safe against a large number of attacks. Due to the page limit, a detailed formal analysis of SDPC using BAN logic can be found at [40].

G. Scyther Implementation Results

Although BAN logic provides a foundation for the formal analysis of security protocols, a few attacks can be undetectable even with BAN logic [36]. However, the critical analysis of BAN-logic in [36] is based on usage of asymmetric cryptography, whereas SDPC utilizes symmetric cryptography. Furthermore, [36] argues that BAN-logic methodology is faulty because it is assumed that physical security and administration do not suffer from the loss of messages by the underlying communication facility or because of host crashes. Owing to replication of contents across the network, in ICN, this assumption has minor effect. Still, for the additional proof of the strength of the SDPC protocol suite, we implemented SDPC, EU-RE [18], and MCAC [17] in an automated security protocol analysis tool, Scyther [21].

TABLE I
Scyther Tool Parameter Settings

Parameter	Settings
Number of runs	1 to 3
Matching type	Find all types of flaws
Search pruning	Find all attacks
Number of patterns per claim	10

We considered four claims: 1) aliveness, 2) weak agreement, 3) noninjective agreement, and 4) noninjective synchronization [20]. These four claims are proven to be true for SDPC by using BAN logic. In Scyther, a protocol is modeled as an exchange of messages among different participating roles. For instance, in NDN-SDPC, the consumer and publisher are in the roles of initiator (I) and responder (R), respectively, whereas the subscription manger is in the role of a server (S). In EU-RE, the publisher acts both as a responder and as a server (R_S), whereas in MCAC, the consumer and publisher are in the roles of I and R, respectively, whereas the third party authenticator has the role of S. The Scyther tool integrates the authentication properties into the protocol specification as a claim event. We tested SDPC, MCAC, and EU-RE by employing the claims mentioned earlier, with the parameter settings given in Table I.

The Scyther results are shown in Table II. It is clear that SubP qualifies all of the protocol claims and no attacks were found. Consequently, for a large number of systems and scenarios, SDPC guarantees safety against a large number of known attacks such as impersonating, man-in-middle, and replay attacks. However, in EU-RE, the author made an assumption that the content provider behaves correctly, i.e., it does not distribute any private contents or decryption rights to unauthorized users; this assumption falsifies the protocol claims, which means EU-RE is weak against several attacks. The Scyther implementation shows that initiator fails to confirm claims 2, 3, and 4.

In MCAC, the TCB along with encrypted communication between routers provides strong security against man-in-middle attack; however, during the bootstrapping, the session key establishment is conducted by the Diffie–Hellman (DH) key distribution algorithm [37] without using a proper authentication procedure. Since the DH algorithm does not inherently provide authentication, it can be secure only if it is properly integrated with another authentication protocol. This weak link in MCAC makes it vulnerable to several attacks, even a man-in-middle attack could be possible if an intruder tempered the session key distribution procedure during bootstrapping process. Therefore, from Scyther implementation results, it can be seen that MCAC fails to qualify a signal claim; further, if we assume the DH key exchange protocol is integrated with an authentication protocol or bootstrap process is hidden from the intruder, then MCAC qualifies all the claims. The inclusion of the authentication

TABLE II
SCYTHER RESULTS FOR SDPC, MCAC, AND EU-RE

Claims	**MCAC [17]**			**MCAC [17] auth.**			**EU-RE [18]**		**NDN-SDPC**		
	I	R	S	I	R	S	I	R_S	I	R	S
Aliveness	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
Weak Agreement	N	N	N	Y	Y	Y	N	Y	Y	Y	Y
Non-injective Agreement	N	N	N	Y	Y	Y	N	Y	Y	Y	Y
Non-injective Synchronization	N	N	N	Y	Y	Y	N	Y	Y	Y	Y

N = Protocol claim is not fulfilled; Y = Protocol claim is fulfilled

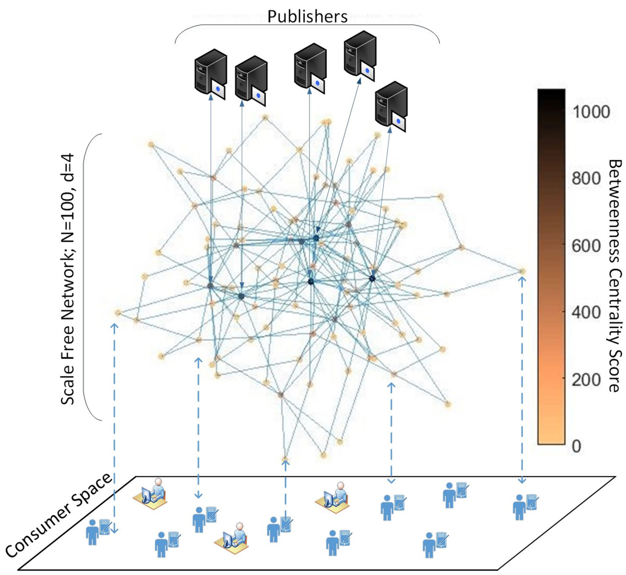


Fig. 4. Network setup for performance evaluation.

process causes the extra processing burden only during the bootstrap process and can be ignored for the next steps in the protocol.

VI. PERFORMANCE EVALUATION

We consider a scale-free network of 200 cache nodes generated using the BarabsiAlbert (BA) model, as shown in Fig. 4, which connects the publisher and the consumer space. Each cache router has a static request routing table. Further, we assume five content publishers in the network. Each publisher has 100 000 secure content items, and a Zipf-distribution with a popularity distribution exponent $\alpha = 0.7$ is used to determine the population of content items in the entire network.

To ensure quick dissemination of the contents in the network, the publishers are connected to the cache routers with the highest betweenness centrality score, which helps bringing system in a steady state in short time. Furthermore, 25 gateway cache routers are connected to the consumer space with a large number of consumers. At any given time, 400–500 consumers are subscribed with each publisher and thus the total number of consumers subscribed to five publishers varies between 2000 and 2500. The size of each content item is 1 GB, which is divided into 10 segments, and the link capacity between two cache routers is 1 Gbps, Finally, least frequently and recently used (LFRU) [7] is used in the experiment as a content replacement scheme.

We implemented the network setup, as described above, in MATLAB and compared the performance of NDN-SDPC against MCAC, EU-RE, and native NDN for two scenarios, 1) using end-to-end encryption, which makes the caching ineffective, as discussed in Fig. 1, and 2) enabling the caching with a conventional way of a shared group key [8], [9]. In Scenario 2, the shared group key enables in-network caching, but the shared group key is unfeasible because the authorized consumers need to keep a large number of keys for effective cache utilization in highly dynamic environments. Moreover, extra decision operations are required to select a proper key and to determine the timing of key deletion. For simplicity, in scenario 2, we only consider the computational and message complexity required to ensure backward and forward secrecy. The processing required to select an appropriate key on the consumer side is ignored. Further, scenario 2 is simulated for different levels of dynamicity in the consumer space, by considering 5, 15, and 25 leave and join requests per unit time, representing cases 1, 2, and 3, respectively. This comparison is made in terms of average download time,[4] publisher load,[5] and timeout interest ratio.[6]

Fig. 5 shows the comparison of the average download time observed when each of the 25 gateway cache routers receives the requests that are generated by a Poisson distribution with a rate $\lambda_j^d = 100$ req/s. The results are considered for different cache sizes of 200 MB to 100 GB; further, in case of MCAC, it is considered that 20% of contents are labeled as h level and 80% as d level.[7] From Fig. 5, it can be seen that NDN with SDPC outperforms EU-RE and native NDN both in scenarios 1 and 2. The performance of NDN in scenario 2 degrades further with the increase in dynamicity of the consumer space. The performance results of EU-RE are interesting; for smaller cache size (200 MB–1 GB), the performance of EU-RE is very close to NDN-SDPC, and it outperforms native NDN both in scenarios 1 and 2. However, the performance gap increases with the increase of the cache size, and it falls down below NDN in scenario 2 with case 3. In Eu-RE, the key revocation and content version are not correlated, and this can be one of the reasons of such performance degradation. For a large cache size, MCAC performs better than EU-RE and NDN in scenario 2 with cases 2

[4]The average download time is defined as the ratio of the total number of requests observed on all 25 cache routers to the time taken to receive all the requested contents at the gateway routers.

[5]The publisher load is defined as the percentage of interests reached at publisher. A high publisher load implies a low cache hit.

[6]The timeout interest ratio is defined as the percentage of interests timed out and retransmitted.

[7]The existence of different levels of content impacts the overall performance of MCAC, as shown in Fig. 6.

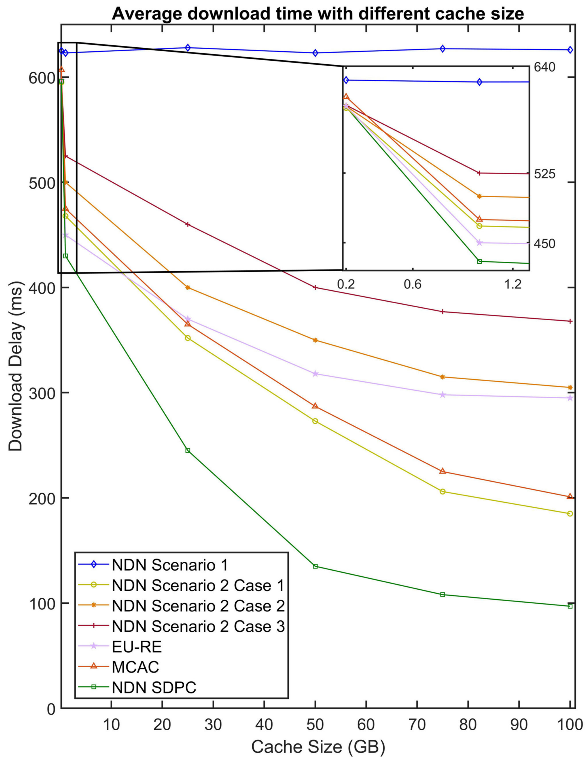


Fig. 5. Average download delay at gateway-cache router.

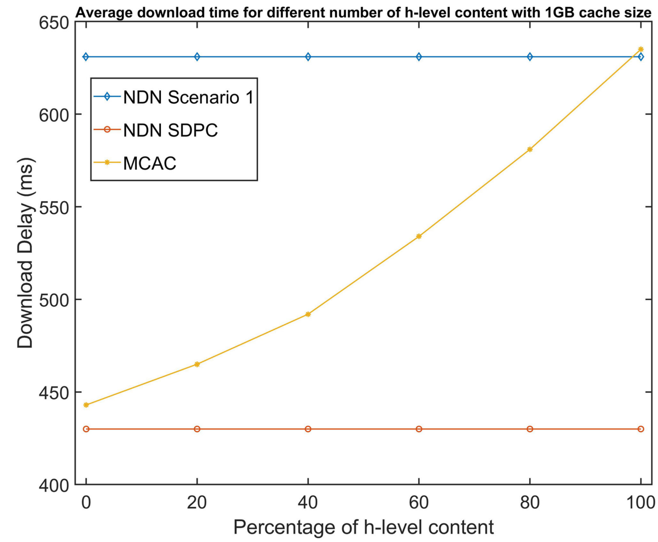


Fig. 6. Average download delay for different numbers of h-level contents.

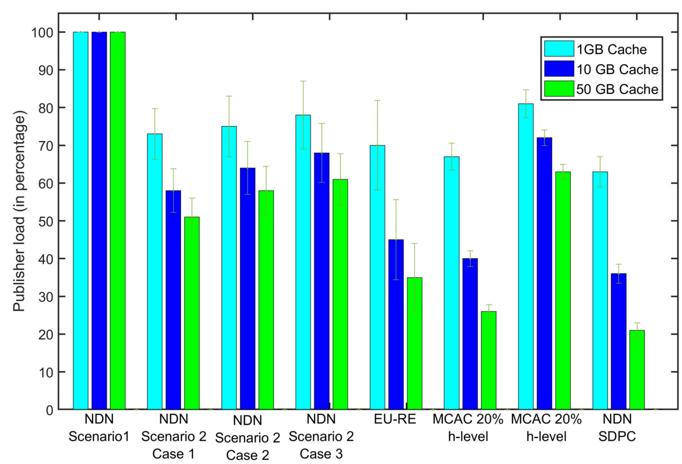


Fig. 7. Publisher load for different cache sizes.

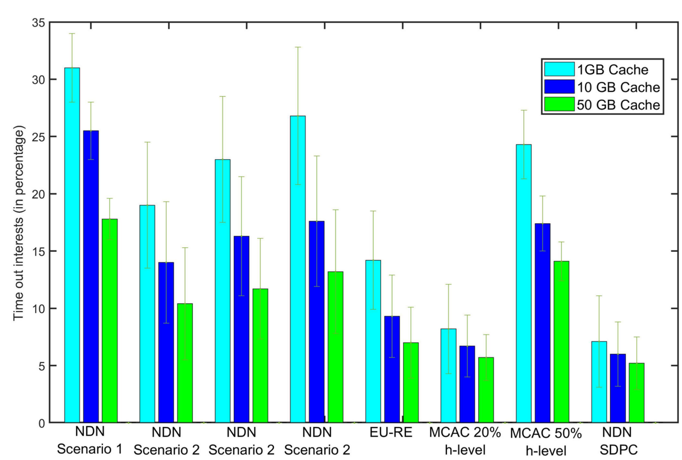


Fig. 8. Timeout interest ratio for different cache sizes.

and 3; however, NDN-SDPC and NDN in scenario 2 with case 1 outperform the MCAC. As discussed earlier, MCAC enforces intermediate routers to implement TCB, which includes several operations and encryption/decryption process for the h-level and the n-level secure content; these extra operations introduce processing delay at intermediate routers, and the performance of MCAC further decreases with increasing the number of h-level contents.

Fig. 6 the comparison of average download time comparison between NDN, NDN-SDPC, and MCAC, for different numbers of h-level contents ranging from 0% to 100% of total traffic, with the fixed 1 GB cache size. From Fig. 6, it is clear that performance of MCAC degrades with increasing number of h-level content. The performance degradation of MCAC with increasing the number of the h-level contents is quite obvious, because h-level contents require no caching policy; hence, all interest packets traverse to the publisher.

Fig. 7 shows the comparison of publisher load. We considered the case-3 level dynamicity of consumer space for EU-RE, MCAC, and NDN-SDPC. From Fig. 7, it is evident that in NDN-SDPC, the publisher load is 12%–20% lower than EU-RE; however, publisher load at MCAC is almost the same as NDN-SDPC, but MCAC's load increases with increasing number of h-level contents. This also implies that NDN-SDPC has higher cache hit ratio.

Similarly, from Fig. 8, it can be seen that EU-RE, NDN-SDPC, and MCAC, with small numbers of h-level contents, suffer with lower number of timeout interest packets. However, NDN-SDPC and MCAC with 20% of h-level contents suffer 35%–50% less than EU-RE; however, this performance metric also shows that performance of MCAC reduces with increasing number of h-level contents. This also implies that in comparison to EU-RE and MCAC, the NDN-SDPC provides better cache diversity.

VII. Conclusion

For effective caching and access control of the protected content in ICN, we proposed an SDPC. The SDPC's keying protocol suite empowers the publisher and consumer to generate multiple symmetric encryption keys with the exchange of a single commitment key. Moreover, SDPC's subscription and content access protocol suite ensures that only authenticated users can acquire the respective key generation information for the requested content. Another important aspect of proposed scheme is the hybrid naming scheme, which provides privacy protection and deters the time analysis attack. The commitment key in SDPC is generated with the publisher's public key, along with other secret credentials, and thus allows the consumer to implicitly verify the originality of the published content. In other words, self-certifying is achieved with the symmetric key cryptography, which makes SDPC free from the expensive computation overhead problem incurred in public key algorithms [24]. Consequently, we believe that the adaptation of SDPC can make NDN more feasible for resource-limited networks such as Internet of Things (IoT), which is one of our future works.

References

[1] C. Ge, Z. Sun, and N. Wang, "A survey of power-saving techniques on data centers and content delivery networks," *IEEE Commun. Surv. Tut.*, vol. 15, no. 3, pp. 1334–1354, 3rd Quarter 2013.

[2] A. Malatras, "State-of-the-art survey on P2P overlay networks in pervasive computing environments," *J. Netw. Comput. Appl.*, vol. 55, pp. 1–23, Sep. 2015.

[3] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proc. ACM CoNEXT'09*, Rome, Italy, Dec. 2009.

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[5] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Comput. Netw.*, vol. 57, no. 16, pp. 3128–3141, Nov. 2013.

[6] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-centric mobile caching network frameworks and caching optimization: A survey," *EURASIP J. Wireless Commun. Netw.*, 2017, Art no. 33, doi: 10.1186/s13638-017-0806-6.

[7] M. Bilal and S. Kang, "A cache management scheme for efficient content eviction and replication in cache networks," *IEEE Access*, vol. 5, pp. 1692–1701, Feb. 2017.

[8] M. Bilal and S. Kang, "A secure key agreement protocol for dynamic group," *Cluster Comput.*, vol. 20, no. 3, pp. 2779–2792, Sep. 2017.

[9] Q. Zhang, Y. Gan, and Q. Zhang, "A dynamic and cross-domain authentication asymmetric group key agreement in telemedicine application," *IEEE Access*, vol. 6, pp. 24064–24074, Jan. 2018.

[10] E. Cho, J. Shin, J. Choi, T. Kwon, and Y. Choi, "A tradeoff between caching efficiency and data protection for video services in CCN," in *Proc. Network Distrib. Syst. Secur. Secur. Emerg. Netw. Technol.*, San Diego, CA, USA, Feb. 2014.

[11] T. Xiaobin, J. Liguo, Z. Zifei, and Y. Pei, "Copyright protection scheme for information-centric networking base on the linear network coding," in *Proc. Chin. Control Conf.*, Chengdu, China, Jul. 2016, pp. 6867–6872.

[12] M. Bilal and S. Kang, "Network-coding approach for information-centric networking," *IEEE Syst. J.*, vol. 13, no. 2, Jun. 2019, doi: 10.1109/JSYST.2018.2862913.

[13] S. Misra, R. Tourani, F. Natividad, T. Mick, N. Majd, and H. Huang, "AccConF: An access control framework for leveraging in-network cached data in the ICN-Enabled wireless edge," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 5–17, Jan. 2019.

[14] S. Misra, R. Tourani, and N. Majd, "Secure content delivery in information-centric networks: Design implementation and analyses," in *Proc. ACM SIGCOMM'13 ICN Workshop*, Hong Kong, China, Aug. 2013, pp. 73–78.

[15] C. Wood and E. Uzun, "Flexible end-to-end content security in CCN," in *Proc. IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, Jan. 2014, pp. 858–865.

[16] Q. Zheng, G. Wang, R. Ravindran, and A. Azgin "Achieving secure and scalable data access control in information-centric networking," in *Proc. IEEE Int. Conf. Commun.*, London, U.K., Jun. 2015, pp. 5367–5373.

[17] Q. Li, R. Sandhu, X. Zhang, and M. Xu, "Mandatory content access control for privacy protection in information centric networks," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 5, pp. 494–506, Sep. 2017.

[18] E. Mannes, C. Maziero, L. Lassance, and F. Borges, "Optimized access control enforcement over encrypted content in information-centric networks," in *Proc. Int. Symp. Comput. Commun.*, Larnaca, Cyprus, Jul. 2015, pp. 924–929.

[19] S. Chow, J. Weng, Y. Yang, and R. Deng, "Efficient unidirectional proxy re-encryption," in *Proc. AFRICACRYPT'10*, Stellenbosch, South Africa, May 2010, pp. 316–332.

[20] C. Cremers, S Mauw, and E. Vink, "Injective synchronisation: An extension of the authentication hierarchy," *Theor. Comput. Sci.*, vol. 367, no. 1–2, pp. 1396–161, Nov. 2006.

[21] C. Cremers and S. Mauw, "Security properties," in *Springer Operational Semantics and Verification of Security Protocols*. Berlin Heidelberg, Germany: Springer-Verlag.

[22] M. Bilal and S. Kang, "Time-assisted authentication protocol," *Int. J. Commun. Syst.*, vol. 30, no. 15, Oct. 2017, Art. no. e3309.

[23] M. Bilal and S. Kang, S. Kim, and J. Park, "Method and apparatus for providing time-assisted authentication protocol," U.S. Patent 20170134369A1, May 11, 2017.

[24] K. Koyama, U. Maurer, T. Okamoto, and S. Vanstone, "New public-key schemes based on elliptic curves over the ring Zn," in *Proc. CRYPTO'91*, Santa Barbara, CA, USA, Aug. 1991.

[25] Y. Wong, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Commun. Surv. Tut.*, vol. 8, no. 2, pp. 2–23, 2nd Quarter 2006.

[26] R. Rivest, "The RC5 encryption algorithm," in *Proc. Conf. Found. Softw. Eng.*, Dec. 1994.

[27] J. Daemen and R. Vincent, AES Proposal: Rijndael. National Institute of Standards and Technology. [Online]. Available: http://csrc.nist.gov/archive/aes/rijndael/Rijndaelammended.pdf

[28] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and C. Hal, Twofish: A 128-Bit Block Cipher. [Online]. Available: https://www.schneier.com/academic/paperfiles/paper-twofishpaper.pdf

[29] M. Burrows and R. Abadi, "A logic of authentication," *ACM Trans. Comput. Syst*, vol. 8, no. 1, pp. 18–36, Feb. 1990.

[30] M. Jakobsson and S. Stamm, "Web camouflage: Protecting your clients from browser-sniffing attacks," *IEEE Secur. Privacy*, vol. 5, no. 6, pp. 16–24, Nov. 2017.

[31] E. AbdAllah, H. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *IEEE Commun. Surveys Tut.*, vol. 17, no. 3, pp. 1441–1454, 3rd Quarter 2015.

[32] R. Klump and M. Kwiatkowski, "Distributed IP watchlist generation for intrusion detection in the electrical smart grid," *Crit. Infrastructure Protection IV*, vol. 342, pp. 113–126, Mar. 2010.

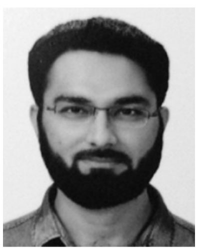
[33] K. Wang, Y. Zhao, S. Liu, and X. Tong, "On the urgency of implementing interest NACK into CCN: From the perspective of countering advanced interest flooding attacks," *IET Netw.*, vol. 7, no. 3, pp. 136–140, May. 2018.

[34] Y. Xin, Y. Li, W. Wang, W. Li, and X. Chen, "A novel interest flooding attacks detection and countermeasure scheme in NDN," in *Proc. IEEE Global Commun. Conf.*, Washington, DC, USA, Dec. 2016, pp. 1–7.

[35] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda, "Privacy risks in named data networking: What is the cost of performance," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 5, pp. 54–57, Oct. 2012.

[36] D. Nessett, "A critique of the Burrows, Abadi and Needham logic," *ACM SIGOPS Operating Syst. Rev.*, vol. 24, no. 2, pp. 35–38, Apr. 1990.

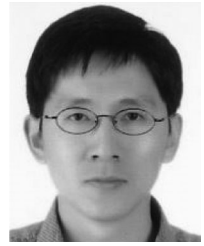
[37] R. Merkle, "Secure communications over insecure channels," *Commun. ACM*, vol. 21, no. 4, pp. 294–299, Apr. 1978.
[38] M. Frash, M. Turkanovic, S. Kumari, and M. Holbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment," *Ad Hoc Netw.*, vol. 36, no. 1, pp. 152–176, Jan. 2016.
[39] K. Mahmood, S. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. Sanaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.
[40] M. Bilal and S. Pack, "Appendix: Secure distribution of protected content in information-centric networking," doi: 10.13140/RG.2.2.21166.36164/1.
[41] M. Bilal, S. Kang, and S. Pack, "Effective caching for the secure content distribution in information-centric networking," in *Proc. IEEE VTC'18 Spring RAFNET Workshop*, Porto, Portugal, Jun. 2018.



Muhammad Bilal (M'16) received the B.Sc. degree in computer systems engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2008, the M.S. degree in computer engineering from the Chosun University, Gwangju, South Korea, in 2012, and the Ph.D. degree in information and communication network engineering from the School of Electronics and Telecommunications Research Institute (ETRI), Korea University of Science and Technology, in 2017.

He is an Assistant Professor with the Division of Computer and Electronic Systems Engineering, Hankuk University of Foreign Studies, Yongin, South Korea. Prior to joining Hankuk University of Foreign Studies, he was a Postdoctoral Research Fellow at Smart Quantum Communication Center, Korea University, Seoul, South Korea, in 2017. His research interests include design and analysis of network protocols, network architecture, network security, IoT, named data networking, BlockChain, cryptology and future Internet.

Dr. Bilal has served as a reviewer of various international journals including IEEE Communications Magazine, IEEE Systems Journal, IEEE Access, IEEE Communications Letters, IEEE Transactions on Network and Service Management Journal of Network, IEEE IoT Journal, IEEE Transactions on Network Science and Engineering and Computer Applications, *Personal and Ubiquitous Computing*, and *International Journal of Communication Systems*. He has also served as a Program Committee Member on many international conferences including IEEE VTC, IEEE IWCMC AI-IOT, and IEEE CCNC. He is an editor of IEEE Future Directions Ethics and Policy in Technology Newsletter and IEEE Internet Policy Newsletter.



Sangheon Pack received the B.S. and Ph.D. degrees from Seoul National University, Seoul, South Korea, in 2000 and 2005, respectively.

In 2007, he joined the Faculty of Korea University, Seoul, Korea, where he is currently a Professor with the School of Electrical Engineering. His research interests include future Internet, softwarized networking (SDN/NFV), information-centric networking (ICN)/delay-tolerant networking (DTN), and vehicular networks.

Dr. Pack is the recipient of IEEE/Institute of Electronics and Information Engineers (IEIE) Joint Award for IT Young Engineers Award 2017, Korean Institute of Information Scientists and Engineers (KIISE) Young Information Scientist Award 2017, Korean Institute of Communications and Information Sciences (KICS) Haedong Young Scholar Award 2013, and IEEE ComSoc APB Outstanding Young Researcher Award. He served as a TPC Vice-Chair for information systems of IEEE WCNC 2020, a Track Chair of IEEE CCNC 2019, a TPC Chair of EAI Qshine 2016, a Publication Co-Chair of IEEE INFOCOM 2014 and ACM MobiHoc 2015, a Co-Chair of IEEE VTC 2010-Fall transportation track, and a publicity Co-Chair of IEEE SECON 2012. He is an Editor of IEEE Internet of Things (IoT) Journal, *Journal of Communications Networks (JCN)*, and *IET Communications*, and he is a Guest Editor of IEEE Transactions on Emerging Topics in Computing (TETC).