Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.4, Aug. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.4.839

분산 환경에서의 효율적인 콘텐츠 인증 기술*

김 대 엽†‡
수원대학교

A Efficient Contents Verification Scheme for Distributed Networking/Data Store*

DaeYoub Kim†‡
Suwon University

요 약

인터넷을 이용하여 콘텐츠 서비스를 운영하려 할 때, P2P, CDN, ICN과 같이 콘텐츠 원배포자에게 집중되는 콘텐츠 요청을 분산 처리하는 기술이 일반적으로 고려되고 있다. 즉, 콘텐츠를 다른 노드들에 분산 저장시킨 후, 분산 저장된 콘텐츠를 이용하여 콘텐츠 요청을 원배포자 뿐만 아니라 여러 노드들이 처리할 수 있게 한다. 그러나 이와 같은 분산 처리 방법은 사용자가 콘텐츠를 수신할 때, 콘텐츠의 원배포자와 전송자가 다를 수 있다는 문제점을 갖고 있다. 즉, 악의적인 사용자에 의하여 콘텐츠가 위/변조된 상태로 전송될 수 있고, 이로 인하여 사용자는 다양한 위험에 노출될 수 있다. 이와 같은 문제를 해결하기 위하여 수신된 콘텐츠를 인증한 후 사용할 것을 권면하지만, 이는 네트워크 전송량 증가 및 서비스 지연의 원인이 될 수 있다. 본 논문에서는 분산 환경에서 보다 효율적으로 콘텐츠를 인증할 수 있는 기술을 제안하고 그 성능을 분석한다.

ABSTRACT

To seamlessly provide content through the Internet, it is generally considered to use distributed processing for content requests converged on original content providers like P2P, CDN, and ICN. That is, after other nodes temporally save content, they handle content requests instead of original content providers. However, in this case, it may be possible that a content sender is different from the original provider of the content. In this case, users may be exposed to various risks. To solve such a problem, it is highly recommended to verify received contents before using them, but it can cause network traffic increases as well as a serious service delay. This paper proposes an efficient content verification scheme for distributed networking/data store environments and analyzes its performance.

Keywords: Futur Internet, NDN, Web Caching, Content Verification, MHT

I. 서 론

초기 인터넷의 주된 목적은 원격 호스트들 간의 안전한 연결을 제공하는 것이었기 때문에, 현재와 같이 다양한 서비스/어플리케이션에서 인터넷을 활용하게 될 것으로 예상하지 않았다. 그러므로 초기 인터넷 개발자들은 현재 인터넷을 기반으로 운영하는 다양한 서비스들이 당면하고 있는 많은 문제점들에 대한 대응 방안을 고려하지 않았다. 이 때문에 빈번한 대용량 콘텐츠 요청 및 전송으로 인한 네트워크 병목 현상, 네트워크 패킷 인증 부재인한 취약점 발생, 사용자 기기의 이동성 증가로 인한 비효율성 증가와 같은

접수일(2015년 5월 30일), 게재확정일(2015년 6월 22일)
* 본 연구는 한국연구재단 연구과제(NRF-2013R1A1A 2 008389) 지원으로 수행하였습니다.
† 주저자, daeyoub69@suwon.ac.kr
‡ 교신저자, daeyoub69@suwon.ac.kr(Corresponding author)

다양한 문제들이 계속적으로 이슈가 되고 있다 [1].

또한, 인터넷을 이용하여 콘텐츠를 사용자에게 제공하거나, 사용자들 간에 콘텐츠를 공유할 수 있도록 하는 다양한 서비스와 어플리케이션이 개발되면서, 네트워크와 사용자 단말기의 상황에 관계없이 지속적인 서비스 (Seamless Service)를 제공하기 위한 기술들 또한 지속적으로 소개되고 있다. 이와 같은 지속적인 서비스를 제공하게 하는 기술 중 하나가 콘텐츠의 원배포자 뿐만 아니라 네트워크 노드나 프락시 시스템 (Multimedia Proxy System)에 콘텐츠를 임시 저장한 후, 이들 네트워크 노드와 프락시 시스템이 콘텐츠의 요청을 원배포자를 대신하여 처리하게 하는 기술이다. 이와 같이 콘텐츠 요청을 분산 처리하는 경우, 콘텐츠 원배포자에게 집중되는 콘텐츠 요청으로 인해 발생하는 네트워크 병목 현상을 해결할 수 있을 뿐만 아니라 콘텐츠 원배포자가 네트워크에 연결되어 있지 않은 상황에서도 사용자는 서비스를 지속적으로 받을 수 있다. 피어 투 피어 네트워킹 (Peer-to-Peer Networking)과 CDN (Content Delivery Network) 기술 등도 이와 같은 측면에서 네트워크의 효율성을 연구한 기술들이라 할 수 있다 [2,3].

다양한 미래 인터넷 기술들도 분산 기술을 이용하여 네크워크의 효율성을 높이려 하고 있다. 특히, Named Data Networking Architecture (NDN)의 경우, 별도의 프락시 시스템을 운영하지 않고 라우터와 같은 네트워크 노드들에 캐쉬 기능을 구현하여 라우터를 통해 중계되는 콘텐트를 임시 저장한 후, 이렇게 저장된 콘텐츠에 대한 요청을 라우터가 수신하면 원배포자를 대신하여 라우터가 해당 요청에 응답하도록 설계 되었다[4][5][6].

그러나 이와 같은 분산 네트워킹 및 분산 데이터 스토리지를 콘텐츠 서비스에 이용할 때, 사용자가 수신한 콘텐츠의 실제 전송자와 원배포자와 다를 수 있다. 이와 같은 상황이 악용될 경우, 악의적인 사용자가 콘텐츠를 위/변조하여 제공하는 시나리오를 쉽게 예상할 수 있다. 특히, 기존 콘텐츠에 악성 코드를 포함시켜 제공하는 경우, 이와 같이 악의적으로 변조된 콘텐츠를 수신한 사용자는 다양한 위험에 노출될 수 있다. 그러므로 분산 환경에서의 콘텐츠를 전송하는 기술은 수신된 콘텐츠를 인증한 후, 해당 콘텐츠를 이용할 것을 사용자들에게 권고하고 있다. 특히, NDN의 경우, 그림 1에서와 같이 전송되는 모든 콘텐츠 (Data Packet)에 원배포자의 전자서명을 첨부하도록 강제함으로써 콘텐츠의 위/변조로 인한 위험을 사전에 탐지할 수 있도록 설계 되었다.

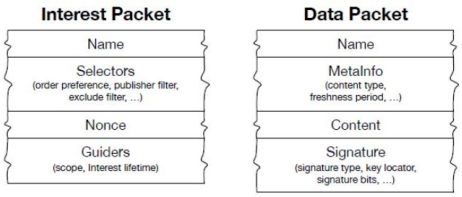

Fig. 1. NDN Message Formats

그러나 이와 같은 콘텐츠 인증 프로세스는 인증 정보를 추가로 전송해야 하기 때문에 일정 수준 네트워크 전송량을 증가 시킨다. 또한, 일반적으로 분산 환경에서는 효율적인 전송을 위하여 콘텐츠를 단편화하여 관리/전송한다. NDN 또한 모든 콘텐츠를 단편화하여 관리/전송한다. 그러므로 단편화된 콘텐츠를 반복해서 인증할 경우, 전체적인 서비스 지연 요인이 될 수 있다.

본 논문에서는 분산 환경에서 콘텐츠 인증 시 발생하는 이와 같은 문제점들을 해결하기 위하여 개선된 콘텐츠 인증 기법을 제안한다. 특히, NDN에서 제안하는 기술을 분석하고, 단편화된 콘텐츠들을 반복하여 인증할 때 요구되는 계산량을 분석하고, 이를 최소화 할 수 있도록 설계 했다. 또한, 제안하는 인증 기술의 성능을 평가하기 위해 기존의 인증 기술들과 성능을 비교 평가하였다.

II. NDN 콘텐츠 검증 기술

2.1 NDN Architecture

그림 2는 NDN의 네트워크 노드가 콘텐츠 요청 메시지 (Interest) 및 응답 메시지 (Data)를 처리하는 과정을 설명 한다. (A~F)는 Interest 처리 절차를, (G~J)는 Data 처리 절차를 각각 설명 한다:

(A) 네트워크 노드의 인터페이스 (Face) 0을 통하여 Interest가 수신된다.

(B) 수신된 Interest에 대응되는 콘텐츠가 CS에 저장되어 있는지 우선 확인한다. 만약 요청된 콘텐츠가 CS에 저장되어 있다면, 저장되어 있는 콘텐츠를 Data로 선택하여 Face 0을 통해 전송한 후, Interest 전송 절차를 종료한다.

(C) 수신된 Interest에 대응되는 Data가 CS에

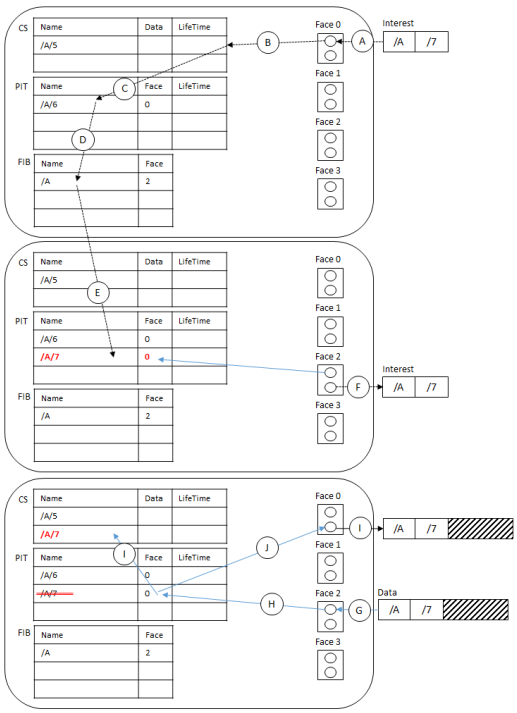

Fig. 2. NDN Interest/Data Processing

저장되어 있지 않다면, 수신된 Interest정보가 PIT (Pending Interest Table)에 기록되어 있는지 확인한다. 만약 해당 기록이 PIT에 있다면, 검색된 PIT 정보의 incoming Face 필드에 Face 0을 추가한 후, Interest 전송 절차를 종료한다.

(D) 수신된 Interest 정보가 PIT에 기록되어 있지 않다면, FIB (Forwarding Information based) 테이블을 참조해서, 수신된 Interest를 전송할 Face (ex. Face 2)를 선택한다.

(E) PIT에 수신된 Interest와 incoming Face 정보 Face 0를 기록한다.

(F) FIB에서 선택한 Face 2를 통하여 수신된 Interest를 전송한다.

(G) Face 2를 통하여 Data가 수신된다.

(H) 수신된 Data에 대응되는 Interest 정보가 PIT에 있는지 확인한다. 만약 없으면, 해당 Data는 폐기처리 된다.

(I) 수신된 Data에 대응하는 Interest 정보가 존재하면, CS에 Data를 저장한다.

(J) Data를 대응되는 Interest 정보의 incoming Face들을 통해서 전송한다. PIT에서 해당 정보를 삭제한다.

그러나 이와 같은 중간 네트워크 노드에 의한 콘텐츠 배포/전송 기술은 콘텐츠가 원배포자가 아닌 불특정 다수의 노드로부터 전송될 수 있기 때문에 콘텐츠 수신자가 콘텐츠 송신자를 정확하게 확인할 수 없다. 그러므로 수신된 콘텐츠가 인증되지 않은 노드로부터 전송된 악성 콘텐츠일 가능성이 내재되어 있다. 이와 같은 문제를 해결하기 위하여 콘텐츠 최종 수신자는 수신된 콘텐츠를 이용하기 전에 반드시 실제 콘텐츠 원생산자/원배포자에 의해서 생성/배포 된 콘텐츠 인지를 확인해야 한다.

또한, 대용량 콘텐츠의 효과적인 배포를 위하여 NDN은 콘텐츠를 단편화(Fragmentation)하여 세그먼트(Segment) 단위로 관리/전송하며, 콘텐츠 인증 역시 segment 단위로 수행한다. 그러므로 대용량 콘텐츠 배포 시, segment 마다 반복적으로 수행되는 인증 절차는 서비스 지연(Service Delay)을 발생시키는 주요 원인이 된다. 그러므로 NDN을 실제 구현하기 위해서는 효율적으로 콘텐츠 인증 기술에 대한 연구가 반드시 필요하다.

2.2 MHT 기반 NDN 콘텐츠 인증 기술

이와 같은 문제를 해결하기 위하여 NDN은 Merkel Hash Tree 기반의 콘텐츠 인증 기술(MHT)을 사용 한다 [4][7]. 그러나 MHT 구현을 위해서는 계층화된 해쉬 값들은 안전하게 계산하기 위하여, 해쉬 값 리스트 전송/처리가 추가적으로 요구되기 때문에 계산 및 전송 오버헤드의 개선이 필요하다.

그림 3은 [4]에서 제안된 MHT 기반 segment 인증 방법의 예이다. 콘텐츠 인증을 위해 콘텐츠 원생성자는 다음과 같은 절차를 따라 NDN segment들을 생성 한다:

(A) 생성자는 배포하려고 하는 콘텐츠를 N ($N \leq 2^n$) 개의 segment들로 단편화 한다: $\{S_0,\cdots,S_{N-1}\}$.

(B) 각각의 segment S_i를 인증하기 위하여 2^n 개의 최하위 노드(Leaf Node)들로 구성된 이진트리(Binary Tree)를 생성한다. 이진트리의 최상위 노드(Root Node)를 N_1이라 하자. 인덱스 순서에 따라 S_i를 이진트리의 최하위 노드 N_{2^n+i}에 할당한 후, S_i의 해쉬 값을 계산하여 N_{2^n+i}의 노드 값

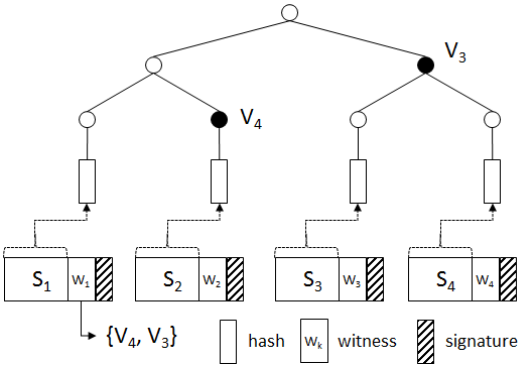

Fig. 3. MHT-based Content Verification

(V_{2^n+i})으로 부여한다:

$$V_{2^n+i} = H(S_i). \quad (1)$$

여기서, $H()$는 단방향 해쉬 함수를 의미한다.

(C) 최하위 노드를 제외한 모든 상위 노드 N_k $(1 \leq k \leq 2^n - 1)$의 노드 값 V_k를 다음과 같이 계산 한다:

$$V_k = H(V_{2k} \| V_{2k+1}). \quad (2)$$

여기서, V_{2k}와 V_{2k+1}은 N_k의 자식 노드 (Child Node)들의 노드 값들을 의미한다. 이와 같은 노드 값 계산 과정을 하위 노드부터 상위 노드 방향으로 반복 수행하여 최상위 노드 N_1의 노드 값 V_1 까지 계산한다.

(D) 생성자는 자신의 전자 서명 키 (SK)를 이용하여 V_1에 대한 전자서명 값을 계산 한다:

$$St = Sign_{SK}(V_1). \quad (3)$$

(E) S_i 검증에 필요한 $witness$ W_i를 생성 한다: W_i는 S_i에 대응하는 최하위 노드 N_{2^n+i} 부터 최상위 노드 N_1 까지 경로(Path)에 포함된 노드들의 형제 노드(Sibling Node)들의 노드 값들로 구성된다. 예를 들어, 8개의 최하위 노드로 구성된 이진트리에서 S_0에 대응하는 최하위 노드 N_8부터 최상위 노드 N_1 까지 경로에 포함된 노드들의 형제 노드들은 N_9, N_5, N_3이다. W_0는 이 형제 노드들의 노드 값 $\{V_9, V_5, V_3\}$으로 구성된다.

(F) S_i 전송을 위한 Data(D_i)를 다음과 같이 구성 하여 배포 한다:

$$D_i = \{S_i, (St, W_i)\}. \quad (4)$$

수신된 콘텐츠가 실제 생성자에 의해 생성된 유효 콘텐츠인지를 검증하기 위하여 콘텐츠 사용자는 다음과 같은 콘텐츠를 검증한다.

(A) 콘텐츠 사용자는 요청하려는 콘텐츠의 첫 번째 segment가 포함된 D_0를 요청하기 위한 Interest를 생성/전송한다. 사용자가 D_0를 수신하면, D_0의 S_0와 W_0를 이용하여 V_1을 계산한다. 계산된 V_1을 이용하여 D_0에 첨부 된 St의 유효성을 검증한다. 만약 St가 유효하면, S_0도 유효하다고 간주한 후, V_1을 임시 저장한 한다.

(B) D_i $(i > 0)$를 요청하기 위한 Interest를 차례로 생성/전송한다. D_i를 수신하면, 수신된 D_i에 포함된 S_i와 W_i를 이용하여 V_1을 계산한다. 계산된 V_1과 앞서 임시 저장한 V_1을 비교한다. 만약 두 값이 같으면 수신자는 S_i가 유효하다고 간주한다.

(C) 콘텐츠의 모든 D_i들이 유효하면, 해당 콘텐츠를 유효하다고 간주하고, 수신된 S_i들을 조합하여 콘텐츠를 재구성한다.

MHT를 대용량 콘텐츠 검증에 적용할 경우, 각각의 S_i 마다 W_i를 추가로 전송해야 하고, V_1을 계산하기 위하여 해쉬 값을 반복적으로 계산해야 한다. 이와 같은 인증 방법은 여전히 전체 서비스를 지연시키는 원인이 될 수 있다.

그림 4는 콘텐츠를 안드로이드 폰과 NDN을 통해 전송할 때 콘텐츠 인증 절차로 인한 서비스 지연 정도를 측정한 결과를 나타낸다. 실험 결과에서 보이듯이 MHT 기반 콘텐츠 인증 기술이 적용된 경우, 콘텐츠 처리 시간이 평균 20% 이상 지연되는 것을 알 수 있다.

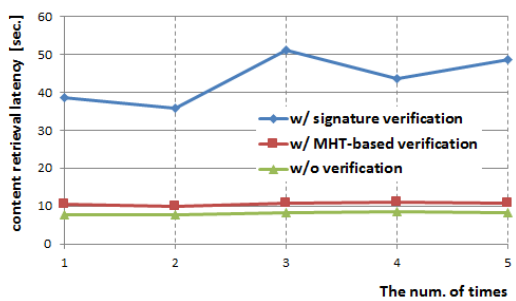

Fig. 4. Verification Overheads

2.3 해쉬 체인 기반 콘텐츠 인증 기술

그림 5는 해쉬 체인을 이용한 segment 인증 시술 (Hash Chain-based Segment Verification, HCSV)을 설명한다. 콘텐츠가 N개의 segment $S_1,\cdots,S_N$로 구성되었다고 가정하자. 콘텐츠의 segment 구성 정보를 갖는 추가적인 segment S_0를 생성한다. j번째 segment 전송을 위한 Data D_j $(0 \leqq j \leqq N)$는 다음과 같이 구성한다:

$$D_j = \begin{cases} S_0 \| H(S_1) \| St_0 \\ S_i \| H(S_{i+1}) \| St_i, i < N \\ S_N \| St_N \end{cases} \quad (5)$$

이 때, D_i는 D_N부터 역순으로 생성하며, St_i는 $i < N$ 이면 $S_i \| H(S_{i+1})$에 대한 전자 서명을, $i = N$ 이면 S_N에 대한 전자 서명을 의미한다. 특히, St_0는 필수 요소이지만, $i > 0$인 St_i는 선택 요소이다.

사용자가 D_i를 수신하면, 해당 사용자가 이전에 D_{i-1}을 검증 했는지 확인한다. 만약 D_{i-1}을 검증하지 않았다면, D_i에 첨부된 St_i을 이용하여 $S_i \| H(S_{i+1})$을 검증한 후, 서명 값이 인증되면 S_i가 검증된 것으로 간주한다. 만약 D_{i-1}을 이미 검증했다면, D_i에 첨부된 $S_i \| H(S_{i+1})$의 해쉬 값을 계산한 후, 저장된 $H(S_i)$와 비교하여 두 값이 같다면 S_i가 검증된 것으로 간주한다. D_i가 검증되면 $H(S_{i+1})$을 임시 저장 한다. 이와 같은 검증 절차를 D_0 부터 D_N 까지 반복하여 수행한다.

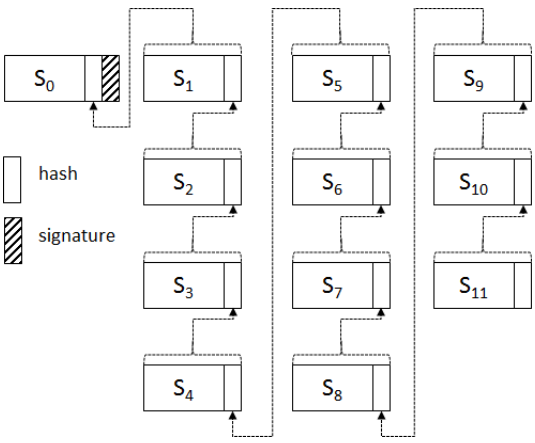

Fig. 5. Hash-chain 기반 Verification

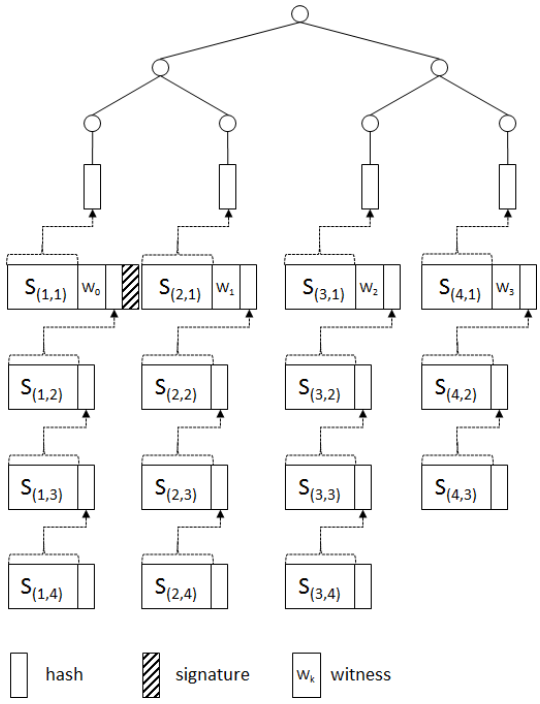

Fig. 6. MHT/Hash-Chain

HCSV를 이용하여 N개의 segment로 단편화된 콘텐츠를 검증하는 경우, 선택 요소인 St_i가 포함되지 않았다면, 추가적인 전송 오버헤드는 $N-1$개의 해쉬 값과 1개의 서명 값만이 추가 전송되며, 콘텐츠 검증을 위하여 N번의 해쉬 값 계산과 1번의 전자서명 값 검증만을 추가적으로 요구 한다. 그러나 HCSV에서 기본적으로 St_0만 전송되는 경우, S_i를 검증하기 위해서 S_{i-1}의 검증이 선행되어야만 한다. 그러므로 S_{i-1} 수신 및 검증에 실패하면, S_i부터는 정상적으로 처리할 수 없다.

이와 같은 segment 손실에 따른 인증 절차의 지연/중단 문제를 해결하기 위하여, segment 마다 전자서명을 첨부하는 방법 외에 그림 6과 같이 MHT와 연동하는 방안(M-HCSV)이 제안되었다 [8][9]. 그러나 전자의 방안은 전체적인 전송량을 30% 정도 증가 시키는 문제점을 갖고 있고, 후자의 개선안 역시 segment 손실에 따른 문제점을 해결하지 못했다.

III. 이중 MHT 기반 HCSV

3.1 D-HCSV

그림 7은 segment 손실 문제를 해결하기 위하여 본 논문에서 제안하는 이중화된 MHT 구조를 적용한 콘텐츠 인증 기술(D-HCSV)을 설명한다. 콘텐츠 segment는 다음과 같이 생성 한다:

(A) 콘텐츠 생성자는 배포하려고 하는 콘텐츠를 R ($R \le 2^r$) 개의 segment들로 단편화 한다: $\{S_0,\cdots,S_{R-1}\}$. 단편화된 segment들을 순서에 따라 N 개씩 묶어, M 개의 segment 그룹들을 생성한다. 이 때, $R \le N \times M$ 을 만족한다. 그림 7은 콘텐츠를 64개의 segment로 단편화 한 후, segment 인덱스 순서에 따라 8개씩 그룹화 하여 8개의 그룹을 생성한 상황을 가정한다. $s[i,j]$는 $k=8i+j$ $(0 \le i,j < 8)$번째 segment S_k를 의미한다.

(B) $s[i,j]$와 함께 전송될 해쉬 값 $H_{[k,1]}$와 $H_{[k,2]}$을 다음과 같이 생성 한다:

$$H_{[k,1]} = h(S_p), \quad p = k+1. \tag{6}$$

$$H_{[k,2]} = h(S_p), \quad p \equiv k+N \mod N \times M. \tag{7}$$

(C) M 개의 최하위 노드로 구성된 이진트리를 생성한다. $s[i,0]$의 해쉬 값을 계산하여, i의 순서에 따라 생성된 이진트리의 최하위 노드 값으로 할당한다. MHT의 노드 값 계산 절차를 따라, 생성된 이진트리의 모든 노드 값 (V_t)들을 계산한다. 최상위 노드 값에 서명하여 $sign_1$을 생성한다.

(D) 각각의 $s[i,0]$에 대하여 서명 검증에 필요한 witness $w[i,1]$을 생성하여, $s[i,0]$에 $sign_1$과 함께 첨부한다.

(E) N 개의 최하위 노드들로 구성된 이진트리를 생성한다. $s[0,j]$의 해쉬 값을 계산하여, i의 순서에 따라 생성된 이진트리의 최하위 노드 값으로 할당한다. MHT의 노드 값 계산 절차를 따라, 생성된 이진트리의 모든 노드 값(v_t)들을 계산한 후, 최상위 노드 값에 서명하여 $sign_2$을 생성한다.

(F) 각각의 $s[0,j]$에 대하여 서명 검증에 필요한 witness $w[j,2]$를 생성하여, $s[0,j]$에 $sign_2$와 함께 첨부한다.

k번째 segment 전송을 위한 Data D_k

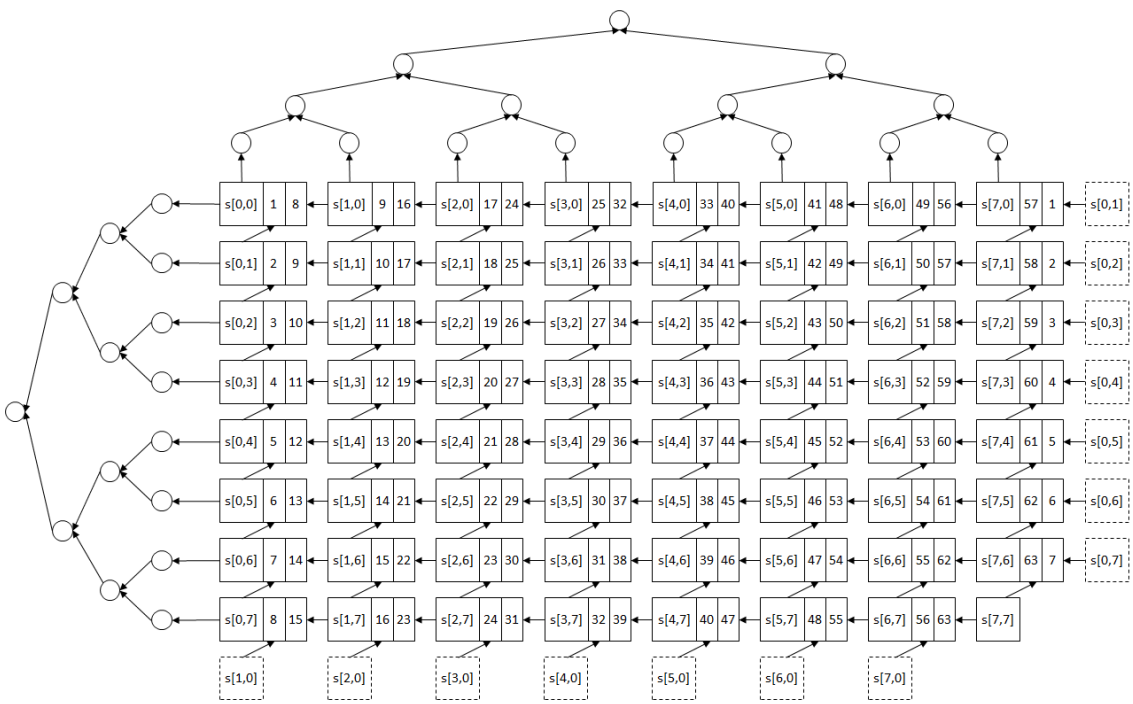

Fig. 7. Content verification using D-Hash Chain

$(0 \leq k < R)$는 다음과 같이 구성 한다:

$$D_k = \begin{cases} S_k \| A_1 \| sign_1 \| A_2 \| sign_2 & , k = 0 \\ S_k \| A_1 \| sign_1 \| H_2 & , i > 0, j = 0 \\ S_k \| H_1 \| A_2 \| sign_2 & , i = 0, j > 0 \\ S_k \| H_1 \| H_2 & , 0 < i, j < R-1 \\ S_k & , k = R-1 \end{cases} \quad (8)$$

여기서 H_1와 H_2는 S_k에 첨부된 $H_{[k,1]}$와 $H_{[k,2]}$를 각각 의미한다. 또한, A_1는 H_1와 $w[i,1]$를, A_2는 H_2와 $w[j,2]$를 각각 의미한다.

사용자가 D_k를 수신하면, 다음과 같이 수신된 segment를 인증 한다: 인증을 위해 H_1을 저장할 메모리 1 개와 H_2를 저장한 메모리 N개를 준비한다. H_1을 저장할 메모리를 m_0라고 하고, H_2를 저장할 N 개의 메모리를 순서에 따라 m_x라 하자 $(1 \leq x \leq N)$.

(A) $k=0$이면, MHT의 인증 절차를 따라 첨부된 $sign_1$을 검증한 후, A_1으로부터 계산된 최상위 노드 값 V_1과 A_2로부터 계산된 최상위 노드 값 v_1을 저장한다.

(B) $i>0$이고, $j=0$ 인 경우, 만약 S_{k-1} 검증이 완료 되었다면, S_k의 해쉬 값을 계산하여 m_0에 저장된 값과 비교한다. 그렇지 않으면, 첨부된 A_1로부터 최상위 노드 값을 계산한 후, 저장된 V_1과 비교한다. 인증이 완료되면, 첨부된 H_1을 m_0에 저장하고, H_2를 m_1에 저장한다.

(C) $i=0$이고, $j>0$이면, 첨부된 A_2로부터 최상위 노드 값을 계산한 후, 저장된 v_1과 비교한다. 두 값이 같으면, 첨부된 H_1을 m_0에 저장하고, H_2를 m_{j+1}에 저장한다.

(D) 그 외의 경우이면, S_k의 해쉬 값을 계산하여 m_0와 m_{j+1}에 저장된 값들과 비교한다. 인증이 완료되면, 첨부된 H_1을 m_0에 저장하고, H_2를 m_{j+1}에 저장한다. 단, S_k가 마지막 segment이면 H_1과 H_2를 저장할 필요는 없으며, 이 경우, 사용자는 콘텐츠가 인증된 것으로 간주한다.

3.2 성능 분석

객관적인 분석을 위해, M-HCSV의 각각의 그룹에 속한 segment는 순차적으로 요청하고 응답 처리 후, 다음 segment를 요청하지만, 그룹의 첫 번째 segment는 별도로 인증이 가능하기 때문에 이전 그룹의 segment 수신 여부와 관계없이 요청할 수 있다고 가정한다.

전체 packet의 수를 R 이라 하고, N 개로 구성된 M 개의 segment 그룹들로 M-HCSV를 구성했다고 가정하자. 패킷 손실 비율/확률을 p라 하고, Interest 전송 후, 수신을 기다리는 최대 대기 시간을 t_1, 평균 응답 시간을 t_2라고 하면, 1개의 segment를 요청 후, 수신하는데 까지 소요될 기대 시간(T)은 다음과 같은 조건을 만족 한다:

$$T \leq (1-p)t_2 + p(\frac{1}{N}t_2 + \frac{N-1}{N}t_1) . \quad (9)$$

이는 그룹 내에서의 segment의 순서에 따라 segment 요청 및 패킷 손실 처리 절차가 다르기 때문이다. 즉,

(1) segment 그룹 내에의 마지막에 위치한 패킷이 중간에 소실되면, 그 다음 패킷은 다음 그룹의 첫 번째 segment이기 때문에 MHT로 인증이 가능하다. 그러므로 재요청이 완료될 때까지 기다리지 않고 segment를 요청하여 처리할 수 있다. 그러므로 Interest를 재요청하기 위해 필요한 시간 이외의 지연 시간이 요구되지 않는다.

(2) 그 외의 경우, 최대 지연 시간은 처리 시간은 segment를 요청한 후, 응답을 기다리는 대기 시간 t_1만큼이 증가한다.

D-HCSV는 segment에 대한 이중 인증 정보를 제공하기 때문에, $S_k = s[i,j]$가 손실이 되어도 그 다음 segment S_{k+1}을 인증할 추가 정보가 제공된다. 그러므로 S_k의 수신 및 처리 완료 시점까지 대기할 필요가 없다. 실제로 S_{k+1}을 인증하기 위해서는 S_k 또는 S_{k+1-M} 중에 하나만 인증 완료 되어 있으면 충분하다. 이 경우, S_k와 S_{k+1-M}가 함께 손실 될 확률은 $P = p^2 \times (N-1)^{-1}$ 이므로, 콘텐츠 요청에 대한 응답 기대시간 T는 다음과 같다:

$$T \leq (1-P)t_2 + P((\frac{1}{M} + \frac{N-1}{N})t_1 + \frac{1}{N}t_2) . \quad (10)$$

UDP의 경우 WLAN 환경에서 1,500 바이트의

패킷을 전송할 경우, 0.8% 정도의 패킷 손실 비율(Packet Loss Rate)이 보고되었다. 특히, 패킷 손실 비율은 전송 패킷의 크기에 비례하여 증가하는 결과를 보였다 [10]. 이와 같은 가정 하에, 시뮬레이션을 위해 1 mega 바이트의 콘텐츠를 CCN을 통해 다운로드 받는다고 가정하고, 현재 CCN의 segment 기본 크기인 4K 바이트로 콘텐츠를 단편화하여 256개의 segment를 생성/전송할 때, 2%의 패킷 손실이 발생하는 상황을 가정하였다. 단, Interest의 패킷 손실은 동일한 조건과 결과를 나타내기 때문에 배제한다. Interest 전송 후, 응답 대기 시간은 3초로 설정하였다. 즉, 대기 시간 내에 Data가 수신되지 않으면 Interest를 다시 전송하다. 그림 8은 이와 같은 가정 아래에서 콘텐츠 요청에 따른 응답/처리 시간을 분석한 결과이다.

성능 분석은 segment가 지연 전송되었을 때를 가정하여 진행하였으며, segment가 영구 손실 되어 전송되지 않을 경우, H-HCSV는 segment의 그룹 내 위치에 따라 인증할 수 없는 segment의 수가 결정된다. 반면, D-HCSV는 이중 인증 구조를 갖고 있기 때문에, 영구 손실되어도 손실된 segment를 제외한 나머지 segment들은 계속해서 인증할 수 있다. 즉, 오류 확산 (Error Propagation) 제어에도 보다 효과적이다.

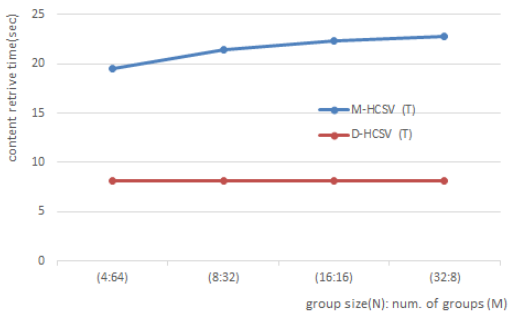

Fig. 8. Content Retrieve Time

IV. 결 론

본 논문에서는 분산 네트워크/스토리지 환경에서 안전한 콘텐츠 배포를 위해 필요한 콘텐츠 인증 기술에 대한 성능 개선 방안을 제안하였다. 특히, 미래 인터넷 기술로 주목 받고 있는 NDN에서 제안하고 있는 콘텐츠 인증 기술을 분석하고, 기존 기술의 문제점을 개선하는데 주력하였다. 본 논문의 결과는 다음과 같은 세 가지 의미를 갖는다. 첫째, 제안된 콘텐츠 인증 방안은 NDN에서 기존에 구현한 MHT의 전송 및 계산 오버헤드를 개선함으로써 NDN 구현 시 주요 delay 요소인 콘텐츠 인증의 효율성을 높일 수 있게 하였다.

둘째, 일반적인 해쉬 체인을 이용한 인증 기술이 갖고 있는 문제점인 중간 인증 정보 손실에 따른 전체 콘텐츠 인증 지연/실패 문제를 해결하기 위하여 이중 해쉬 체인 기법을 적용하였다. D-HCSV는 손실에 따른 오류 범위를 해당 segment로 제한할 수 있기 때문에, 스트리밍 서비스 등에 적용 시 M-HCSV에 비해 양질의 서비스를 제공할 수 있다.

셋째, 인증 정보를 이중화함으로써 일반적인 해쉬 체인을 이용하여 콘텐츠를 인증할 때보다 콘텐츠를 위/변조하기 더욱 어렵도록 설계하였다.

References

[1] M. Y. Chen, E. Kıcıman, E. Fratkin, A. Fox and E. Brewer, "Pinpoint: Problem Determination in Large, Dynamic Internet Services," Proceedings of the International Conference on Dependable Systems and Networks (DSN'2), pp.595-604, June 23-26, 2002.

[2] L. Wang, "Content, Topology and Cooperation in In-network Caching," Series of Publications A Report A-2015-1, Department of Computer Science, University of Helsinki, Finland, 2015.

[3] M. Arifuzzaman, K. Yu and T. Sato, "Collaboration between Network Playersof Information Centric Network: An Engineering-Economic Analysis," Journal of ICT, Vol. 2, pp. 201 - 220, 2015.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," ACMCoNext, pp. 1-6, Dec. 2009.

[5] J. Pan, S. Paul, and R. Jain, "A Survey of the Research on Future Internet Architectures," IEEE Communications Magazine, pp. 26-36, July, 2011.

〔6〕 L. Zhang, et al., "Named Data Networking (NDN) Project," NDN-0001, Oct. 2010.
〔7〕 R. Merkle, "Protocol for public key cryptosystems," IEEE Sympo. Research in Security and Privacy, Apr.1980.
〔8〕 S. Hyun, P. Ning, A. Liu and W. Du, "Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Network," Proc. Int′,l Conf. Information Processing in Sensor Networks (IPSN ′,08), pp. 445-456, 2008
〔9〕 D. Y. Kim, J. S. Park, "Efficient Contents Verification Scheme for Contents-Centric-Networking," The Journal of Korean Institute of Comm. and Inform. Sciences, vol. 39, no. 4, pp. 234-241, April, 2014.
〔10〕 Xylomenos, G., Polyzos, G.C., "TCP and UDP performance over a wireless LAN," INFOCOM 99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. vol. 2, pp. 439 - 446, Mar. 1999.

〈저자소개〉

김 대 엽 (DaeYoub Kim) 종신회원
2000년 2월: 고려대학교 수학과 박사
2000년 3월: 텔리맨 CAS 팀 책임 연구원
2002년 8월: 시큐아이 정보보호연구소 책임 연구원
2012년 2월: 삼성전자 종합기술원 수석 연구원
2012년 3월~현재: 수원대학교 정보보호학과 조교수
〈관심분야〉 DRM/CAS, 난독화, 스마트카드, 미래인터넷 보안