http://dx.doi.org/10.7840/kics.2014.39B.4.234

콘텐츠 중심 네트워킹을 위한 콘텐츠 인증 기술

김 대 엽◆°, 박 재 성*

Efficient Contents Verification Scheme for Contents-Centric-Networking

DaeYoub Kim◆°, Jaesung Park*

요 약

콘텐츠 제공자 주변 네트워크에서 발생하는 병목 현상으로 인한 네트워크 비효율성을 개선하기 위하여, 콘텐츠/정보 중심 네트워킹 기술은 콘텐츠 전송에 참여한 네트워크 노드들이 해당 콘텐츠를 임시 저장한 후, 해당 콘텐츠에 대한 요청 메시지를 다시 수신하면 앞서 저장된 콘텐츠를 요청자에게 전송한 후, 요청 메시지를 콘텐츠 제공자에게 전달하지 않고 콘텐츠 전송 프로세스를 종료한다. 그러나 이러한 콘텐츠 전송 방식은 수신자가 실제 콘텐츠 제공자를 확인할 수 없기 때문에 수신된 콘텐츠를 이용하기 전에 반드시 콘텐츠를 인증해야만 한다. 그러나 이와 같은 콘텐츠 인증 프로세스는 서비스 운영 지연을 발생시키는 원인 중 하나가 되고 있다. 본 논문에서는 콘텐츠 인증에 따른 문제점을 살펴보고, 효율적인 콘텐츠 인증을 위한 개선된 운영 방안을 제안하고 그 성능을 평가한다.

Key Words : Future Internet, CCN, Web Caching, Content Verification, MHT

ABSTRACT

To improve network inefficiency caused by network congestion around a content-source, content-centric networking (CCN) allows network nodes transmitting contents to temporarily cache received contents and then send back the cached contents if the nodes receive relevant request messages without forwarding the request messages to content-sources. However, because nodes receiving requested contents through CCN cannot recognize real senders of the received contents, the receivers need to verify each received contents before using them. But such a verification procedure can cause long service/operation delay. Hence, in this paper, we review the problem of contents verification, propose an improved verification procedure and evaluate its performance.

Ⅰ. 서 론

현재 인터넷이 직면하고 있는 다양한 문제점들을 해결하고, 인터넷을 통한 멀티미디어 콘텐츠 서비스의 급속한 증가에 효과적으로 대응하기 위하여 다양한 미래 인터넷 기술 연구가 진행되고 있다[1-8]. 특히, 콘텐츠 제공자에게 집중되는 네트워크 트래픽 (Network Traffic)으로 인해 발생되는 네트워크 병목 현상을 해결하고, 대용량 멀티미디어 콘텐츠를 효과적으로 배포하기 위하여 많은 미래 인터넷 기술들은 콘텐츠 전송 시, 다른 사용자들에 의해 이전에 요청되어져서 이미 배포된 콘텐츠들(Content Replica)을 네트워킹에 적극적으로 활용하는 방안을 고려하고 있다. Peer-to-Peer 네트워킹 기술과 Content Delivery Network 기술 등도 이러한 측면에서 연구된 기술 중 하나라 할 수 있다. 미래 인터넷에서는 이와 같은 방안을 보다 일반화

※ 본 연구는 2013년도 정부의 재원으로 한국연구재단의 지원을 받아 수행된 연구 결과임 (No. NRF-2013R1A1A2008389).
◆° First Author and Corresponding Author : Suwon University, Department of Information Security, daeyoub69@suwon.ac.kr, 정회원
* 수원대학교 정보보호학과, jaesungpark@suwon.ac.kr, 종신회원
논문번호 : KICS2014-04-117, Received April 7, 2014; Revised April 17, 2014; Accepted April 17, 2014

하여 네트워킹 아키텍처 레벨에서 이를 구현하려고 시도하고 있다. 미래 인터넷 기술로 제안된 콘텐츠 중심의 네트워킹 (Contents-Centric Networking, CCN)이 대표적인 예이다[4-5].

CCN은 라우터 또는 게이트웨이와 같은 네트워크 노드들에 콘텐츠 임시 저장(Caching) 기능을 구현하고, 이들 노드들이 수신된 콘텐츠를 사용자 또는 다른 노드에게 중개할 때, 해당 콘텐츠를 임시 저장하도록 설계 되었다. 또한, 임시 저장된 콘텐츠에 대한 요청 메시지 (Interest)를 네트워크 노드가 수신하면, 저장하고 있는 콘텐츠를 요청자에게 전송한 후, 수신된 Interest를 더 이상 중개하지 않고 Interest 처리 절차를 종료한다. 그림 1은 CCN의 네트워크 노드가 Interest 및 응답 메시지 (Data)를 처리하는 과정을 설명 한다. (A~F)는 Interest 처리 절차를 설명하고, (G~J)는 Data 처리 절차를 설명 한다.

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

(B) 수신된 Interest에 대응되는 Data가 내부 스토리지 (ContentStore, CS)에 저장되어 있는지 확인한다. 만약 저장되어 있다면, Face 0을 통해 해당 Data를 전송한 후, Interest 처리를 종료한다.

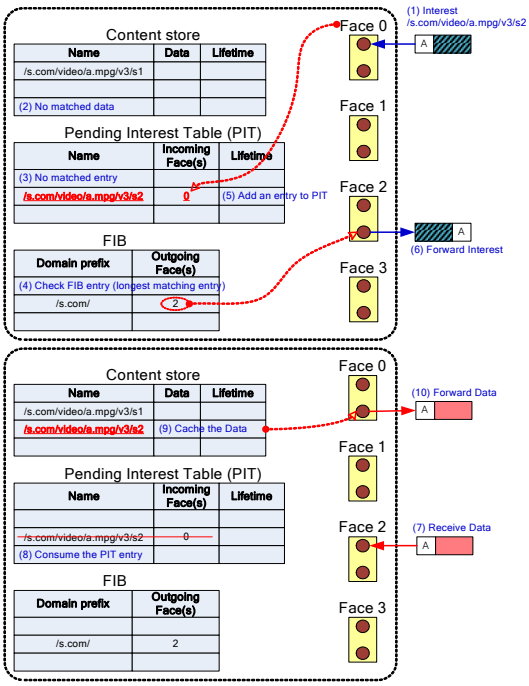

그림 1 CCN Networking 절차
Fig. 1. CCN Networking Procedure using Content Request Message (Interest) and Response Message (Data).

(C) CS에 수신된 Interest에 대응되는 Data가 저장되어 있지 않다면, Interest에 대응 되는 정보가 PIT (Pending Interest Table)에 있는지 확인한다. 만약 있다면, 해당 정보의 incoming Face 필드에 Face 0을 추가한다.

(D) PIT에 대응되는 정보가 없다면, 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.

이와 같이 중간 노드들을 이용한 콘텐츠 배포는 콘텐츠 소스 또는 제공자에게 유입되는 Interest가 네트워크 계층에서 효율적으로 분산 처리되게 함으로써, 콘텐츠 제공자 또는 초기 생성자의 네트워크 주위에서 발생할 수 있는 네트워크 병목 현상을 해결하고 네트워크 효율성을 높일 수 있을 것으로 기대된다.

그러나 이와 같은 중간 노드에 의한 콘텐츠 배포/전송 메커니즘은 메시지 수신자가 메시지 송신자를 확인할 수 없기 때문에 수신된 콘텐츠가 인증되지 않은 노드로부터 전송된 악성 콘텐츠 일 수 있다는 문제점을 갖고 있다. 그러므로 콘텐츠 요청자는 수신된 콘텐츠를 이용하기 전에 반드시 실제 콘텐츠 생산자에 의해서 생성 된 정상적인 콘텐츠 인지를 확인해야 한다. 또한, 대용량 콘텐츠의 효과적인 배포를 위하여 CCN은 콘텐츠를 단편화하여 세그먼트 (segment) 단위로 관리/전송하며, 콘텐츠 인증 역시 segment 단위로 수행한다. 그러므로 대용량 콘텐츠 배포 시 segment 마다 반복적으로 수행되는 인증 절차는 service time delay를 발생시키는 주요 원인이 되고 있다. 그러므로 CCN을 실제 구현하기 위해서는 효율적으로 콘텐츠 인증 기술에 대한 연구가 반드시 필요하다.

이와 같은 문제를 해결하기 위하여 CCN은 기본적으로 Merkel Hash Tree 기반의 콘텐츠 인증 기술 (MHT)을 사용 한다[4,9]. 그러나 MHT의 구현을 위해서는 계층화된 해쉬 값을 인증하기 위하여 해쉬 값 리

스트 전송 및 처리가 추가로 필요하기 때문에 계산 및 전송 오버헤드의 개선이 필요하다.

또한, 무선 센서 네트워크 환경에서 안전하게 코드를 배포하기 위하여 MHT를 기반으로 한 코드 인증 기술 (Code Verification Scheme, CVS)도 제안되었다[10].

본 논문에서는 MHT와 CVS를 소개하고, MHT의 오버헤드를 개선한 해쉬 체인 기반콘텐츠 인증 기법 (Hash Chain-based Content Verification, HC^{2}V)을 제안하고, 기존에 제안된 기술들과 성능을 비교 평가한다. 또한, 제안하는 HC^{2}V를 효율적으로 구현하기 위하여 CVS의 구조를 HC^{2}V에 적용시키기 위한 개선안(MHT-based HC^{2}V, MHC^{2}V)도 함께 제안한 후, 기존에 제안된 기법들과 성능을 비교 평가한다.

Ⅱ. 콘텐츠 인증 기술

2.1 MHT 기반 CCN Segment 생성

안전한 콘텐츠 배포를 위해 콘텐츠 제공자는 다음과 같은 절차를 수행 한다:

(A) 콘텐츠를 일정한 크기로 단편화 하여 N ($N \le 2^n$) 개의 segment $\{S_1,\cdots,S_N\}$을 생성한다.

(B) 생성된 S_i를 효과적으로 관리하기 위하여 2^n 개의 말단 노드 (Leaf Node)로 구성된 이진 트리 (Binary Tree)를 생성한다. 생성된 이진 트리의 최상위 노드 (Root Node)를 N_1이라 하자. 이 때, 각각의 S_i에는 그 순서에 따라 생성된 이진트리의 말단 노드 N_{2^n+i}가 할당된다.

(C) S_i의 해쉬 값을 다음과 같이 계산 한 후, 대응된 N_{2^n+i}의 노드 값 V_{2^n+i}으로 할당 한다:

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

여기서, $H()$는 단방향 해쉬 함수를 의미한다.

(D) 말단 노드를 제외한 상위 노드 N_k의 노드 값 V_k는 다음과 같이 계산 된다:

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

여기서, N_{2k}와 N_{2k+1}은 N_k의 자식 노드 (Child Node)를 의미한다. 이와 같은 노드 값 계산을 하위 노드부터 상위 노드 방향으로 반복해서 수행하여 최상위 노드 값 V_1이 계산될 때까지 수행한다.

(E) 콘텐츠 제공자는 자신의 전자서명 키를 이용하여 V_1에 대한 전자서명 값을 계산 한다: $Sign_{pri} = Sign_{privateKey}(V_1)$. 이 서명 값을 이용하여 S_i를 검증한다.

(F) S_i 검증을 위하여 콘텐츠 제공자는 $witness$ W_i를 다음과 같이 계산 한다: W_i는 S_i에 대응하는 말단 노드 N_{2^n+i} 부터 최상위 노드까지의 경로(Path)에 포함된 노드들의 형제 노드 (Sibling Node)들의 노드 값들로 구성된다. 예를 들어, 8개의 말단 노드로 구성된 이진 트리에서 S_0에 대응하는 W_0는 3개의 노드 값 $\{V_9, V_5, V_3\}$으로 구성된다.

(G) $\{Sign_{pri}, W_i\}$를 S_i와 함께 패키징 해서 전송을 위한 CCN segment CS_i를 생성한 후, CS_i를 배포한다.

2.2 MHT 기반 CCN Segment 검증

수신된 콘텐츠가 실제 제공자에 의해 생성된 유효 콘텐츠인지를 검증하기 위하여 콘텐츠 요청자는 다음과 같은 절차를 수행한다.

(A) 콘텐츠 요청자는 이용하려는 콘텐츠의 첫 번째 CCN segment CS_0에 대응되는 Interest를 전송한다. 요청자가 Data로 CS_0를 수신하면, CS_0에 포함된 S_0와 W_0를 이용하여 앞 절에서 설명한 방법처럼 상위 노드 값들을 반복하여 계산함으로써 V_1을 계산한다. 이렇게 계산된 V_1을 이용하여 CS_0에 첨부 된 $Sign_{pri}$의 유효성을 검증한다. 만약 $Sign_{pri}$이 유효하다면, CS_0가 유효하다고 간주하고 V_1을 임시 저장한 한다.

(B) Interest를 이용하여 CS_i $(i>0)$를 차례로 요청한다. CS_i를 수신하면, 수신된 CS_i에 포함된 S_i와 W_i를 이용하여 V_1을 계산한다. 이렇게 계산된 V_1과 앞서 임시 저장한 V_1을 비교한다. 만약 두 값이 같으면 수신자는 CS_i가 유효하다고 간주한다.

(C) 콘텐츠의 모든 CCN segment들이 유효하면, 해당 콘텐츠를 유효하다고 간주하고, 수신된 S_i들을 재조합하여 콘텐츠를 재구성한다.

S_0 검증 시, 전자서명 검증을 통하여 검증된 V_1을 저장 한 후, 후속 S_i 검증 시, 서명 값을 검증하지 않고 단순히 V_1 값만 비교하면 충분하기 때문에 segment 검증 소요 시간을 보다 효과적으로 줄일 수 있다. 그러나 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 프로세스는 여전히 전체 서비스를 지연시키는 원인이 될 수 있다. 그림 2는 콘텐츠를 안드로이드 폰과 CCN을 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.

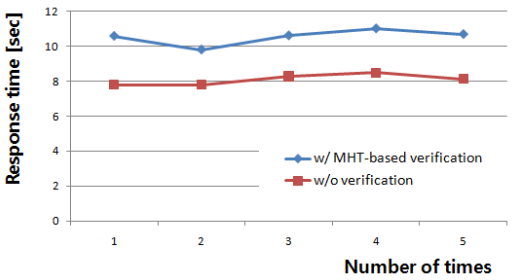

그림 2. MHT 기반 콘텐츠 인증 절차 구현 시 Interest/Data 응답시간 비교 분석
Fig. 2. The comparison result of Interest/Data response time

2.3 MHT 기반 CVS

2.3.1 CVS 코드 검증 데이터 생성 절차

그림 3은 무선 센서 네트워크 환경에서 코드 배포 시, 배포된 코드 인증을 위해 제안된 CVS를 설명 한다. 인증 정보가 포함된 코드는 다음과 같은 절차에 따라 생성 된다:

(A) 전송 될 코드를 L개의 페이지 $(Page_{(1)},...,Page_{(L)})$로 분할 한 후, 각각의 페이지 $Page_{(i)}$를 다음과 같이 N개의 패킷으로 분할하여 관리 한다:

$$Page_{(i)} = \{P_{(i,1)},\cdots,P_{(i,N)}\}. \quad (3)$$

(B) 전송 패킷 $P^h_{(i,j)}$ 을 생성하기 위하여 마지막 페이지 $Page_{(L)}$에 속한 j번째 패킷의 해쉬 값을 계산한 후, $Page_{(L-1)}$의 j번째 패킷에 첨부 한다:

$$P^h_{(L-1,j),} = P_{(L-1,j)} \| H(P_{(L,j)}). \quad (4)$$

(C) 생성 된 $Page_{(L-1)}$의 전송 패킷들

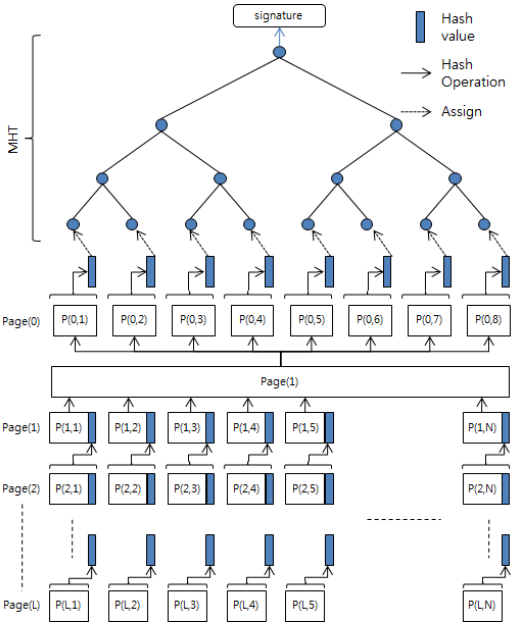

그림 3. MHT-base CVS의 운영 예제
Fig. 3. MHT-base CVS in the case M=8

$\{P^h_{(L-1,j)}\}_{j=1,...,N}$의 해쉬 값을 각각 계산하여 $Page_{(L-2)}$의 대응하는 패킷에 첨부 한다:

$$P^h_{(L-2,j),} = P_{(L-2,j)} \| H(P^h_{(L-1,j)}). \quad (5)$$

(D) 이와 같이, i번째 페이지의 j번째 패킷을 $i-1$번째 페이지의 j번째 패킷에 첨부하는 작업을 $Page_{(2)}$에 포함된 전송 패킷들의 해쉬 값을 $Page_{(1)}$의 패킷에 첨부하여 전송 패킷을 만들 때 까지 반복하여 수행 한다:

$$P^h_{(i-1,j),} = P_{(i-1,j)} \| H(P^h_{(i,j)}). \quad (6)$$

(E) 생성 된 $Page_{(1)}$의 전송 패킷들의 해쉬 값을 계산한 후, 계산된 해쉬 값들을 순서에 따라 연접하여 인증 데이터 h를 생성 한다:

$$h = H(P^h_{(1,1)}) \| \cdots \| H(P^h_{(1,N)}). \quad (7)$$

(F) 생성된 h를 MHT의 말단 노드의 개수 $(M=2^m)$로 분할하여 $Page_{(0)}$를 생성 한다:

$$Page_{(0)} = \{P_{(0,1)},\cdots,P_{(0,M)}\}. \quad (8)$$

(G) $P_{(0,i)}$를 순서에 따라 MHT의 말단 노드에 차례로 할당한다. 본 논문에서는 설명을 간단하게 하기 위하여 $N=M=2^m$이라 가정한다.

(H) II장 1절에서 설명한 것처럼 MHT의 노드 값들을 계산한 후, 최상위 노드 값 V_1에 대한 전자서명 값 $Sign_{pri}$을 생성한다.

(I) $P_{(0,j)}$에 검증에 필요한 W_j를 계산한 후, 다음과 같이 전송 패킷 $P^h_{(0,j)}$을 생성 한다:

$$P^h_{(0,j)} = \{P_{(0,j)}, W_j, Sign_{pri}\}. \quad (9)$$

전송되는 최종 코드는 다음과 같이 생성 된다:

$$\left\{P^h_{(i,j)}\right\}_{i=0,\ldots,L;j=1,\ldots,N} \quad (10)$$

2.3.2 CVS를 이용한 코드 검증 절차

최종 코드를 수신한 수신자는 다음과 같이 코드 검증을 수행 한다:

(A) 수신자는 MHT를 이용하여 $Page_{(0)}$의 모든 패킷들을 검증한다. 검증 결과 유효한 페이지로 판단되면, $Page_{(0)}$의 패킷에 첨부된 패킷들을 순서에 따라 재조립 (reassembling) 절차를 수행하여 인증 데이터 h를 생성한 후, 생성된 h를 임시 저장한다.

(B) $Page_{(1)}$을 검증하기 위하여 $Page_{(1)}$에 포함된 각각의 패킷 $P^h_{(1,j)}$에 대한 해쉬 값을 계산한 후, 단계 (A)에서 저장한 h의 j번째 해쉬 값과 비교한다. 만약 두 값이 같으면 $P^h_{(1,j)}$을 유효한 패킷으로 간주하고 $P^h_{(1,j)}$에 첨부된 $h(P^h_{(2,j)})$를 임시 저장 한다.

(C) $1<i<L+1$을 만족하는 모든 i에 대하여, 순서에 따라 $Page_{(i)}$를 검증하기 위하여 $Page_{(i)}$에 포함된 각각의 패킷 $P^h_{(i,j)}$에 대한 해쉬 값을 계산한 후, 이전 단계에서 임시 저장 된 $h(P^h_{(i-1,j)})$과 비교한다. 만약 두 값이 같으면 $P^h_{(i,j)}$을 유효한 패킷으로 간주하고 $P^h_{(i,j)}$에 첨부된 $h(P^h_{(i+1,j)})$를 임시 저장 한다. 단, $i=L$인 경우, $P^h_{(L,j)}$에 $h(P^h_{(L+1,j)})$이 첨부되어 있지 않기 때문에 $h(P^h_{(L+1,j)})$를 임시 저장하지 않는다.

2.3.3 CVS 특성 분석

CVS 적용 시, 다음과 같은 비효율성이 발견 된다:

(A) $Page_{(1)}$부터 $Page_{(L)}$까지 각각의 페이지를 인증하기 위하여, N개의 해쉬 값을 계속 저장/관리해야 된다.

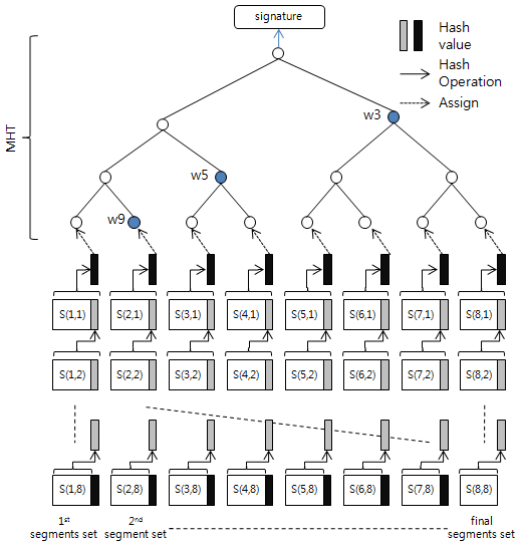

그림 4. MHT-based HC^{2}V 운영 예제
Fig. 4. MHT-based HC^{2}V Procedure Example

(B) 각각의 페이지에 포함된 패킷의 수가 N으로 고정되어 있다. 그러므로 $Page_{(L)}$의 패킷의 수가 N보다 적은 경우, 패딩 패킷을 추가로 $Page_{(L)}$에 포함시켜야 한다.

(C) MHT의 말단 노드의 개수 M이 페이지를 구성하는 패킷의 수 N과 최대 전송 패킷의 크기에 따라 가변적으로 운영된다. 그러므로 MHT의 구성 정보를 추가로 전송해야 된다. 특히, N은 운영시스템 또는 하드웨어 스펙에 따라 다르게 적용될 수 있기 때문에 MHT 구조가 이에 따라 달라 질 수 있으며, 이와 같은 가변성은 MHT를 하드웨어로 구성하기 어렵게 만드는 요인이 된다.

Ⅲ. 해쉬 체인 기반 콘텐츠 검증 기법

CVS는 네트워크를 통하여 전송된 단위 코드를 검증하기 위해 제안되었다. 그러므로 수신된 코드가 $(L+1)$개의 페이지를 모두 포함하고 있는 경우를 고려하여 설계되었다. 그러나 CCN에서 콘텐츠 요청 및 전송은 단편화 된 segment의 순서에 따라 순차적으로 진행된다. 즉, 콘텐츠 요구자는 첫 번째 segment를 요청한 후, 요청한 segment가 수신되면, 순차적으로 다음 segment를 요청한다. 그러므로 CVS를 CCN 콘텐츠 검증에 그대로 적용시킬 경우, N개의 segment를 수신 및 검증이 완료되어야, 다음 N개의 segment를

정상적으로 처리할 수 있다. 또한, CVS가 갖고 있는 비효율성 역시 개선될 필요가 있다.

본 절에서는 콘텐츠 인증 과정에서 발생되는 오버헤드를 개선하기 위하여 CCN의 특성이 반영된 해쉬 체인(Hash Value Chain) 기반의 콘텐츠 검증 기법 (Hash Chain-based Content Verification, HC^{2}V)을 제안 한다.

3.1 HC^{2}V 기법

콘텐츠가 N개의 segment $S_1,\cdots,S_N$로 단편화 된 경우, i번째 CCN segment P_i는 다음과 같이 구성 한다:

$$P_i=\begin{cases} S_1\|H(P_2)\|Sign_{pri}, & \text{if } i=1 \\ S_i\|H(P_{i+1}) & \text{,if } 1<i<N \\ S_N & \text{,if } i=N. \end{cases} \quad (11)$$

이 때, CCN segment는 P_N부터 역순으로 생성하며, $Sign_{pri}$은 $S_1\|H(P_2)$에 대한 콘텐츠 생성자의 전자서명 값을 의미한다.

콘텐츠 요청자는 수신된 P_1에 첨부된 $Sign_{pri}$을 이용하여 $S_1\|H(P_2)$에 대한 검증한 후, $H(P_2)$를 임시 저장 한다. P_2를 요청하고, 수신된 P_2를 검증하기 위하여 P_2의 해쉬 값을 계산한 후, 앞서 저장된 $H(P_2)$ 값과 비교한다. 두 값이 같으면 P_2가 유효하다고 간주하고, $H(P_2)$ 대신 P_2에 첨부된 $H(P_3)$를 임시 저장 한다. 이와 같은 검증 절차를 마지막 segment인 P_N 검증까지 반복하여 수행한다.

HC^{2}V를 이용하여 N개의 segment로 구성된 콘텐츠를 검증하는 경우, 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 HC^{2}V는 i번째 segment를 검증하기 위해서는 반드시 $i-1$번째 segment의 검증이 선행되어야만 한다. 그러므로 i번째 segment 수신 및 검증에 실패하면, $i+1$번째 segment부터는 정상적으로 처리할 수 없다.

3.2 MHT-based HC^{2}V

HC^{2}V의 단점을 개선하기 위하여 본 논문에서는 CVS의 구성을 HC^{2}V에 적용한다. 그림 4는 제안하는 MHT-based HC^{2}V (MHC^{2}V)를 설명한다.

(A) $M=2^m$개의 말단 노드로 구성된 MHT를 생성한다. 말단 노드의 개수는 사용되는 응용 프로그램에 따라 고정된 값을 사용할 수도 있고, 또는 서로 다른 값을 사용할 수도 있다.

(B) 콘텐츠를 분할하여 N개의 segment $\{S_1,\cdots,S_N\}$ 들을 생성한다.

(C) 생성된 S_i들을 index 순서에 따라 M개의 segment 집합 $\{S^1,S^2,\cdots,S^M\}$에 차례에 따라 다음과 같이 할당 한다: 각각의 segment 집합 S^i $(i=1,...,M-1)$는 $n=N/M$ 개의 segment 들로 구성 되고, S^M은 할당되지 않은 나머지 segment들로 구성된다. S^i의 j번째 원소 $S_{(i,j)}$는 다음과 같이 정의 된다:

$$S_{(i,j)}=S_{(i-1)\times n+j}. \quad (12)$$

(D) segment S_k를 $S_k=S_{(i,j)}$라 할 때, 인증 정보를 포함하는 CCN segment는 아래와 같이 생성 된다:

D-1) $S_{(i,1)}$의 해쉬 값 $H(S_{(i,1)})$를 계산 한 후, MHT의 i번째 말단 노드 값으로 할당한다. MHT의 최상위 노드 값 V_1을 계산 한 후, V_1에 대한 제공자의 전자서명 $Sign_{pri}$을 생성하고, $H(S_{(i,1)})$ 검증에 필요한 $witness$ W_i를 생성한다.

D-2) $S_{(i,j)}$ 전송을 위한 CCN segment P_k를 다음과 같이 구성 한다:

$$P_k=\begin{cases} S_k^h\|W_i\|Sign_{pri}, & \text{if } j=1 \\ S_k^h & \text{,if } 1<j<n \\ S_k & \text{,if } j=n \text{ or } k=N \end{cases} \quad (13)$$

여기서, $S_k^h=S_k\|h_{k+1}$, $h_{k+1}=H(P_{k+1})$를 각각 의미한다.

수신된 P_k를 검증하기 위해서 수신자는 인덱스 k의 순서에 따라 다음과 같은 절차를 반복해서 수행한다.

(A) P_k에 포함된 S_k가 $S_{(i,1)}$이면, Ⅱ장 1절에서 설명한 MHT 검증 절차에 따라 S_k를 검증한다. 검증이 완료되면, P_k에 첨부된 h_{k+1}를 임시 보관한다.

(B) P_k에 포함된 segment S_k가 $S_{(i,j)}$, $j>1$이면, $H(P_k)$를 계산 한 후, P_{k-1} 검증 시 저장된 h_k와 계산된 $H(P_k)$ 값을 비교하여, 두 값이 같으면 P_k를 유효한 값으로 간주한다. 이 때 $j<n$이면, P_k에 첨부된 h_{k+1}을 h_k를 대신하여 임시 저장한다.

3.3 성능 평가

표 1은 basic MHT, CVS, 그리고 본 논문에서 제

표 1. 콘텐츠 인증 기술 성능 비교 분석
Table 1. The Performance Evaluation Result of content verification schemes

	Communication overhead (CMO)	Computation Overhead (CO)		Storage Overhead (SO)
		Signer	Verifier	
Basic MHT	$n \times N$	$2N-1$	$(n+1) \times N$	1
CVS	$N+((n_2-1) \times N_2)$	$N+(N_2-1)$	$N+(n_2 \times N_2)$	N_2+1
HC^{2}V	$N-1$	N	N	1
MHC^{2}V	$N+((n_2-1) \times N_2)$	$N+(N_2-1)$	$N+(n_2 \times N_2)$	1

안한 HC^{2}V와 MHC^{2}V의 성능 비교를 나타낸다.

$N=2^n=2^{n_1+n_2}$개의 segment로 구성된 콘텐츠 처리를 가정한다. CVS의 성능을 평가하기 위하여 해당 콘텐츠는 $N_1=2^{n_1}$ 개의 페이지로 구성되어 있으며, 각각의 페이지는 $N_2=2^{n_2}$ 개의 segment로 구성되어 있다고 가정 한다; 또한, 성능 비교를 위하여 MHC^{2}V은 N_2개의 말단 노드로 구성된 MHT를 사용하고, 각각의 segment set S^i는 N_1개의 segment로 구성되어 있다고 가정한다.

CMO, CO, SO는 각각 $witness$ 처리를 위한 전송 오버헤드, 해쉬 처리를 위한 계산 오버헤드, $witness$ 및 서명 처리를 위한 검증자의 메모리 오버헤드를 각각 의미한다.

Basic MHT를 CCN에 적용할 경우, CMO와 CO를 효과적으로 줄일 수 있으나 CVS에 비해 SO가 증가한다. HC^{2}V의 경우 CMO, CO 및 SO를 모두 최소화 할 수 있으나, 중간에 하나의 segment 인증 처리에 실패할 경우, 실패한 segment 뒤에 수신되는 segment를 처리할 수 없다. MHC^{2}V은 CMO와 CO의 성능을 CVS와 동일한 수준으로 개선하면서도 SO가 증가하지 않는다.

Ⅳ. 결 론

본 논문에서는 CCN에서 제안하고 있는 콘텐츠 인증 기술을 분석하고 전송, 계산, 저장 오버헤드를 개선하기 위한 새로운 운영 방안을 제안했다. 본 논문의 결과는 다음과 같은 세 가지 의미를 갖는다. 첫째, 제안된 인증 방안은 저장 오버헤드의 증가 없이 전송 및 계산 오버헤드를 감소시킴으로써 CCN 구현 시 주요 delay 요소인 콘텐츠 인증의 효율성을 높일 수 있게 하였다.

둘째, segment 처리를 위해 segment set을 구성할 때, segment set의 원소 수를 가변적으로 운영할 수 있어 콘텐츠 특성에 따라 유연하여 적용할 수 있다. 마지막으로 콘텐츠 크기에 상관없이 MHT의 크기를 고정시켜 사용할 수 있기 때문에 하드웨어로의 구현이 용이하다.

References

[1] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," *ACM SIGCOMM*, pp. 181-192, Oct. 2007.

[2] B. Ahlgren, et al., "Second NetInf architecture description," 4WARD EU FP7 Project, Deliverable D-6.2 v2.0, Apr. 2010.

[3] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 26-36, Jul. 2011.

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

[5] L. Zhang, et al., "Named data networking (NDN) project," NDN-0001, Oct. 2010.

[6] J. H. Kim and S. K. Kim, "Recent trend on the semantic web and future internet services," in *Proc. KICS*, pp. 316-317, Korea, 2011.

[7] Y. J. Kim, J. S. Park, and B. M. Chin, "Standardization on future internet," in *Proc. KICS*, pp. 342-343, Korea, 2011.

[8] M. K. Park, S. H. Min, B. C. Kim, J. Y. Lee, and D. Y. Kim, "Implementation of a future internet testbed using software based MAC in IP capsulator," in *Proc. KICS*, pp. 240-241,

Korea, 2011.

[9] R. Merkle, “Protocol for public key cryptosystems,” *IEEE Symp. Research in Security and Privacy*, Apr. 1980.

[10] S. Hyun, P. Ning, A. Liu, and W. Du, “Seluge: secure and DoS-Resistant code dissemination in wireless sensor network,” *Int. Conf. Inf. Process. Sensor Netw.*, 2008.

김 대 엽 (DaeYoub Kim)

2000년 2월 : 고려대학교 수학과 박사

2000년 2월~2002년 6월 : 시큐아이 정보보호연구소 차장

2002년 9월~2012년 2월 : 삼성전자 종합기술원 PM

2012년 3월~현재 : 수원대학교 정보보호학과 조교수

<관심분야> 정보보호이론, 미래 인터넷 보안

박 재 성 (Jaesung Park)

1995년 2월 : 연세대학교 전자공학과 졸업

1997년 2월 : 연세대학교 전자공학과 석사

2001년 2월 : 연세대학교 전기, 전자공학과 박사

2001년~2002년 : University of Minnesota (PostDoc.)

2002년~2005년 : LG전자(선임연구원)

2005년 현재 : 수원대학교 정보보호학과 부교수

<관심분야> 네트워크 성능 분석 및 프로토콜 개발