Vol.7, No.9 pp.235~242 pISSN: 2287-5891 https://doi.org/10.3745/KTCCS.2018.7.9.235

The Shortest Authentication Path for Performance Improvement of MHT Contents Authentication Method in Distributed Network Environment

DaeYoub Kim†

ABSTRACT

Various technologies have been developed to more efficiently share content such as P2P, CDN, and CCN. These technologies take a common approach that content request packets is responded by distributed network nodes or hosts, not by a single content distributor. Such approaches not only resolve network congestion around content distributors, but also make it possible to distribute content regardless of the system and network status of content distributors. However, when receiving content from distributed nodes/hosts, not from authenticated distributors, users cannot practically identify which node/host sent content to them. Due to this characteristic, various hacking caused by the malicious modification of content is possible. Therefore, to make such approaches more secure, a content authentication technique is required. In this paper, we propose a improved operation of MHT used in CCN for authenticating distributed content. Then we evaluate the proposed method by comparing its performance with the existing technology.

Keywords : P2P, CDN, CCN/NDN, Data Authentication, Data Integrity, MHT

분산 네트워크 환경에서의 MHT 콘텐츠 인증 기술 성능 개선을 위한 최소 인증 경로에 관한 연구

김 대 엽†

요 약

인터넷을 이용한 콘텐츠 공유를 보다 효율적으로 구현하기 위하여 P2P, CDN, CCN과 같은 다양한 네트워크 기술들이 개발되었다. 이러한 기술들은 공통적으로 콘텐츠 배포자에게 집중되는 콘텐츠 요청 패킷을 네트워크에 분산된 다수의 노드들/호스트들이 처리하도록 설계되어, 네트워크 병목 현상을 해결하고 콘텐츠 배포 시스템이나 네트워크 상태와 상관없이 지속적으로 콘텐츠를 배포할 수 있는 장점을 갖고 있다. 그러나 분산된 노드/호스트로부터 콘텐츠를 전송 받는 경우, 사용자가 실제 콘텐츠 전송 노드/호스트를 식별/인증할 수 없기 때문에 공격자 개입 및 악의적인 콘텐츠 변경을 통한 다양한 해킹 공격에 취약하다. 그러므로 분산 노드/호스트를 이용한 네트워킹 기술의 경우, 콘텐츠 인증 기술은 핵심 기술 요소들 중 하나이다. 본 논문에서는 CCN에 적용된 콘텐츠 인증 기술인 MHT 기반의 콘텐츠 인증 기법을 소개하고, MTH의 인증 정보 중복 전송 문제를 해결하고 전송량을 개선하기 위하여 인증 경로 계층 값을 최소화하는 방안을 제안한다. 또한, 기존 기술들과의 성능 비교를 통하여 개선안의 성능을 평가한다.

키워드 : P2P, CDN, CCN/NDN, 데이터 인증, 데이터 무결성, MHT

1. 서 론

실시간 데이터 및 대용량 콘텐츠를 유/무선 네트워크 기술을 이용하여 사용자에게 전송/공유하는 다양한 서비스가 지속적으로 개발되고 있다. 특히, 모바일 기기의 급속한 보급과 SNS (Social Network Service) 및 클라우드 기반의 콘텐츠 서비스와 같은 다양한 서비스의 이용이 보편화 되면서 이와 같은 콘텐츠 서비스를 이용하기 위한 유/무선 네트워크 데이터 전송이 폭발적으로 증가하고 있다. 그러나 이와 같은 전송 데이터의 증가에 대응하기 위하여 유/무선 통신 선로를 확장하는 것은 서비스 제공자에게 매우 많은 투자비용을 요구할 뿐만 아니라, 물리적 통신 선로 확장 속도에 비하여 데이터 전송량 증가 속도가 훨씬 더 빠르게 증가하는 추세이기 때문에 물리적 해결책뿐만 아니라 네트워크 효율성을 개선할 수 있는 기술적 대안이 추가로 요구된다[1, 2]. 이러한 기술적 대

※ 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A1B03034215).
† 종신회원 : 수원대학교 정보보호학과 조교수
Manuscript Received : May 10, 2018
Accepted : June 7, 2018
* Corresponding Author : DaeYoub Kim(daeyoub69@suwon.ac.kr)

응 방안 중 하나가 P2P (Peer-to-Peer) 네트워킹 기술과 CDN (Content Delivery Network) 서비스라 할 수 있다. P2P/CDN은 사용자가 요청하는 콘텐츠를 해당 콘텐츠의 원배포자 (Content Provider/Publisher, CP) 뿐만아니라 동일한 콘텐츠를 이전에 다운로드 받은 사용자들이나 콘텐츠 프락시 서버 (Content Proxy Server, CPS)와 같이 사용자가 요청한 콘텐츠를 저장하고 있는 여러 호스트/시스템들을 통하여 콘텐츠를 제공 받을 수 있도록 설계되었다. 이러한 네트워킹 기술은 CP로 집중되는 콘텐츠 요청 메시지들을 효과적으로 분산처리 함으로써 네트워크 효율성을 개선할 수 있을 뿐만 아니라, CP의 시스템/네트워크 상태에 상관없이 지속적으로 콘텐츠를 배포할 수 있다[3, 4].

초기 인터넷의 주된 개발 목적은 원격 호스트들 사이의 네트워킹을 안전하게 연결하는 것이었다. 그러므로 현재와 같은 다양한 서비스들이 인터넷을 기반으로 구현될 때 발생하는 여러 요구사항들을 초기 인터넷 설계자들은 고려하지 않았다. 그러므로 이러한 서비스들을 이용할 때 발생되는 급속한 데이터 전송량 증가와 이로 인한 네트워크 병목현상, 호스트 간 인증 기술의 부재와 같은 취약한 보안 구조로 인한 보안 침해 사고, 모바일 기기 증가로 인한 사용자 기기의 빈번한 이동에 따른 비효율성과 같은 다양한 문제점들이 드러나고 있다[5]. 특히, IT 융복합 서비스의 증가와 모바일 기기 및 네트워크에 연결된 센서를 비롯한 다양한 기기의 증가는 이와 같은 인터넷의 문제점들을 더욱 심화시킬 것으로 예상된다. 그러므로 IT 융복합 서비스의 발전 및 저변 확대를 위해서는 이와 같은 인터넷의 구조적인 문제들을 해결해야 한다.

인터넷의 기술적 문제들을 해결하고, 데이터 및 정보를 인터넷을 통하여 보다 안전하고 효율적으로 제공/공유하기 위한 미래 인터넷 기술 및 아키텍처 연구가 활발하게 진행되고 있다[6-8]. 정보 중심 네트워킹 기술(Information Centric Networking, ICN)은 이러한 미래 인터넷 아키텍처 중 대표적인 기술이다. ICN은 CP에게 전송되는 콘텐츠 요청 메시지를 효율적으로 분산 처리하기 위하여 CPS나 네트워크 노드 (Network Node)에 콘텐츠를 저장한 후, CPS/네트워크 노드들이 CP를 대신하여 콘텐츠 요청 메시지에 응답하도록 설계되었다. 특히, ICN 기술 중 하나인 콘텐츠 중심 네트워킹 (Content Centric Networking/Named Data Networking, CCN/NDN)은 네트워크 노드에 구현된 콘텐츠 임시 저장 기능과 콘텐츠의 계층화된 고유 이름을 참조하여 네트워크 패킷을 라우팅 하는 기술을 구현함으로써 네트워크 전송 경로상의 중간 노드들이 콘텐츠 요청 메시지를 직접 처리할 수 있도록 설계되었다[7-10].

기존의 호스트 중심의 네트워킹 기술은 호스트 식별자를 네트워크 패킷에 포함시킴으로써 상대 호스트를 식별하고, 식별된 호스트를 인증하기 위하여 다양한 계층의 인증 기술을 활용하였다. 그러나 ICN은 네트워크 패킷의 전송 효율성을 높이기 위해 P2P/CDN처럼 사용자는 다수의 시스템/노드들로부터 콘텐츠를 제공받게 된다. 그러므로 사용자가 콘텐츠를 수신했을 때, 해당 수신자는 콘텐츠를 제공한 실제 시스템/노드를 식별/인증할 수 없다. 이러한 ICN의 특성이 악의적으로 이용될 경우, 콘텐츠 위/변조가 가능하고, 위/변조된 콘텐츠를 이용한 다양한 공격이 가능할 수 있다. 그러므로 ICN을 이용한 콘텐츠 전송/배포 서비스를 운영하기 위해서는 콘텐츠 인증 기술이 필수적으로 요구된다.

콘텐츠 인증을 위해 CCN은 해당 콘텐츠 생성자 (Content Publisher, CP)의 전자 서명을 콘텐츠에 첨부하도록 강제 규정하고 있다. 대용량 콘텐츠 전송을 지원하기 위해 CCN은 콘텐츠를 일정 크기 이하의 세그먼트들로 단편화한 후, 각각의 세그먼트를 하나의 데이터로 간주하여 처리하고, 콘텐츠를 구성하는 모든 세그먼트들을 개별 인증하도록 제안되었다. 그러나 개별 세그먼트 단위의 콘텐츠 인증 기술을 적용할 경우, 콘텐츠를 구성하는 전체 세그먼트들을 무두 인증하는데 많은 시간이 소요되어 콘텐츠 서비스가 지연될 수 있다. 이와 같은 문제를 개선하기 위해서 CCN은 머클 해쉬 트리 (Merkle Hash Tree, MHT)를 사용하여 콘텐츠 인증 및 개별 콘텐츠 세그먼트 인증을 동시에 수행한다[11, 12].

그러나 [13]에서 MHT 운영 시 발생되는 과도한 중복 연산과 그로인한 비효율성이 지적되었고, [14]에서는 MHT의 인증 정보의 중복 전송으로 인한 비효율성이 지적되었다.

본 논문에서는 [13]과 [14]에서 제안된 MHT의 중복 연산 및 전송을 개선하는 방안의 성능을 향상 시키기 위하여 새로운 MHT 운영 방법을 제안한다. 이를 위하여 본 논문에서는 세그먼트 인증을 위한 인증 경로를 단축시킴으로써 전송 및 연산 효율성을 개선하였다.

2. CCN

CCN을 포함한 ICN의 주요 특징 중 하나는 네트워크 노드와 같이 별도의 시스템 또는 노드에 캐싱된 데이터를 활용하는 것이다. 이와 같은 네트워크 캐시를 효과적으로 활용하기 위하여 CCN은 다음과 같은 두 가지 차별화된 기술을 사용한다:

(1) 콘텐츠 이름(Content Name) 기반의 패킷 라우팅

(2) 전자 서명 기반의 패킷 인증.

전자는 네트워크 노드에 CCN의 데이터 캐싱 기능을 구현할 때, 효율적으로 캐싱 된 데이터를 탐색/관리할 수 있도록 한다. 후자는 캐싱 된 데이터가 사용자에게 전송되었을 때, 사용자가 수신된 데이터를 신뢰하고 이용할 수 있도록 한다 [7].

CCN은 콘텐츠 요청 패킷(Interest) 전송에 의하여 네트워킹 프로세스의 진행이 시작되며, Interest에 대응되는 응답 패킷 (Data)은 Interest가 전송된 경로의 역경로를 따라서 사용자에게 전송된다. 이때, Interest 전송 경로 위에 있는 네트워크 노드는 Interest의 네트워크 계층 정보를 이용하여 요청된 콘텐츠의 캐싱 여부를 확인하고, 라우팅 경로를 결정해야 한다. 그러므로 Interest의 네트워크 계층 정보는 콘텐츠의 식별자 정보를 포함한다. CCN은 특정 원격 호스트로부터 콘텐츠를 전송 받는 메커니즘이 아니기 때문에 특정 호스트와의 네트워크 접속을 위해 필요했던 호스트 식별자의 필요성이 매우 낮다. 그러나 네트워크 경로 상에 캐싱된 데이터가 없다

면, CP에게 Interest가 전송되어야 하므로, Interest의 네트워크 계층 정보에는 CP의 식별자도 포함되어야 한다.

이와 같은 두 가지 특성에 따라, CCN은 IP 주소와 같은 호스트 식별자 대신에 콘텐츠 식별자와 CP 식별자 정보로 구성된 콘텐츠 이름을 네트워크 주소로 활용하고, 콘텐츠 이름을 기반으로 Interest/Data의 라우팅 경로를 결정한다. 또한, 콘텐츠 이름을 참조하여 Interest 패킷을 라우팅하기 위하여 콘텐츠 이름은 계층화하여 구성한다.

데이터의 위/변조 여부를 사용자가 검증 할 수 있도록 전송되는 각각의 Data는 Publisher의 전자 서명을 포함하고 있으며, 사용자는 Data에 첨부된 전자 서명 값을 검증하여 데이터의 Publisher를 인증하고 동시에 데이터 위/변조 여부를 검증한다. CCN의 구성과 실제 운영 방안은 [7, 13]에 자세히 설명되어 있다.

3. MHT 기반 데이터 인증

3.1 MHT 기반 데이터 인증

CCN은 CP 인증과 세그먼트 인증 기능을 제공한다. CP 인증은 수신된 콘텐츠의 CP를 식별하고 인증하는 것을 의미한다. CP 인증을 통해서 수신된 콘텐츠의 신뢰성을 일차적으로 검증할 수 있다.

전송 효율성과 안정성을 고려하여 CCN은 콘텐츠를 세그먼트들로 단편화한 후, 각각의 세그먼트를 독립된 Data로 처리하기 때문에, 만약 수신된 세그먼트가 요청한 콘텐츠의 세그먼트가 아니면, 콘텐츠 전체를 수신 한 후에도 해당 콘텐츠를 정상적으로 이용하지 못할 수도 있다. 그러므로 수신된 세그먼트가 사용자가 요청한 콘텐츠의 단편화된 세그먼트임을 검증하는 세그먼트 인증이 필요하다.

CCN은 MHT 기반의 콘텐츠 인증 기법을 제안하였다 [11-15]. Fig. 1은 MHT 기반의 콘텐츠 인증 절차를 예를 들어 설명한 것이다. CP는 다음과 같은 과정을 수행한다:

(1) 콘텐츠가 일정 크기를 갖는 $N\ (\leq 2^n)$ 개의 세그먼트들로 단편화 되었다면, 2^n개의 리프 노드(Leaf Node)로 구성된 이진 트리(Binary Tree)를 구성한다. 이 경우, 이진트리는 $n+1$개의 계층(Level)으로 구성되며, 리프 노드는 0 계층, 루트 노드는 n 계층이라 한다. Fig. 1은 $N=2^3=8$인 경우를 가정한다. 루트 노드를 $M[1]$이라하고, 상위 계층부터 노드의 순서에 따라 노드 번호를 부여한다. i번째 노드 $M[i]$의 노드 값을 V_i라 하자.

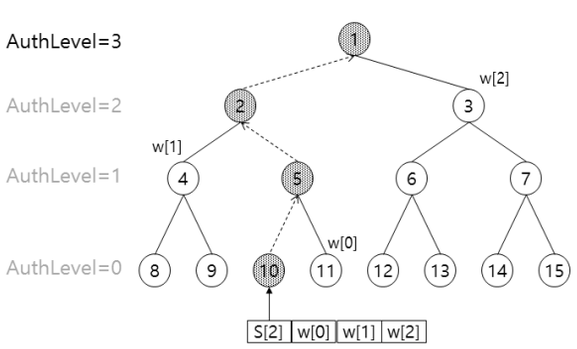


Fig. 1. MHT-based Content Authentication for CCN

(2) 각각의 세그먼트들은 콘텐츠 구성 순서에 따라 생성된 이진 트리의 리프 노드에 할당된다. 개별 세그먼트의 해시 값을 계산 한 후, 해당 세그먼트에 할당된 리프 노드의 노드 값으로 사용한다. 즉, Fig. 1에서 i번째 세그먼트 $S[i]$에 할당된 리프 노드 $M[8+i]$의 노드 값은 $V_{8+i}=H(S[i])$이다.

(3) 리프 노드들을 제외한 k번째 중간/루트 노드 $M[k]$의 노드 값 V_k는 다음과 같다:

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

여기서, V_{2k}와 V_{2k+1}은 $M[k]$의 두 자식 노드들(Child Nodes)의 노드 값들이다. 이와 같은 방법으로 하위 노드들부터 루트 노드까지 각각의 노드 값들을 순차적으로 계산한다.

(4) CP의 전자 서명 키 ($priK$)를 이용하여 루트 노드의 노드 값 V_1에 서명하여, $sign=E_{priK}(V_1)$을 생성한다. 콘텐츠의 세그먼트 $S[i]$와 $sign$을 함께 패키징 하여 Data를 생성한다.

(5) $S[i]$가 포함된 Data를 수신하면, 사용자는 Data에 패키징 된 $sign$을 검증하기 위해 V_1을 계산해야 한다. 이를 위하여 $S[i]$에 대응하는 리프 노드부터 루트 노드까지의 인증 경로에 포함된 노드들의 노드 값을 계산해야 한다. 그러므로 사용자는 인증 경로에 포함된 노드들의 형제 노드(Sibling Node)들의 노드 값들이 필요하다. 사용자에게 V_1 계산에 필요한 정보를 제공하기 위하여 CP는 인증 경로 상의 노드들의 형제 노드들의 노드 값들을 Data에 함께 패키징하다. 이렇게 Data에 추가된 노드 값들($w[0], w[1], ..., w[n-1]$)을 인증 정보(Witness)라 한다.

(6) 사용자가 요청한 콘텐츠에 대응하는 Data를 수신하면, Data에 포함된 세그먼트와 인증정보를 이용하여 리프 노드부터 루트 노드까지의 인증 경로 상의 노드들의 노드 값들을 차례로 계산하여 V_1을 획득한 후, Data에 패키징된 $sign$을 검증한다.

콘텐츠 인증을 보다 효율적으로 수행하기 위하여, CCN은 첫 번째 세그먼트 $S[0]$를 검증 할 때, Data에 패키징된 $sign$이 유효하다면, 계산된 V_1을 저장한다. 이 후, 세그먼트 $S[i]$ $(i>0)$를 검증할 때에는, 수신된 Data로 부터 계산된 V_1과 앞서 $S[0]$ 검증 단계에서 저장한 V_1을 비교하는 것으로 세그먼트 검증을 대신한다. 각각의 세그먼트 검증 시, 매번 $sign$을 검증할 필요가 없으므로 세그먼트 검증에 소요되는 시간을 효과적으로 줄일 수 있다.

3.2 MHT의 해시 값 중복계산으로 인한 비효율성 개선

MHT를 대용량 콘텐츠 배포에 적용할 경우, 각각의 세그먼트 마다 인증 정보를 함께 전송해야 하고, V_1 계산을 위해 하위 노드 값들을 반복적으로 중복해서 계산해야만 한다.

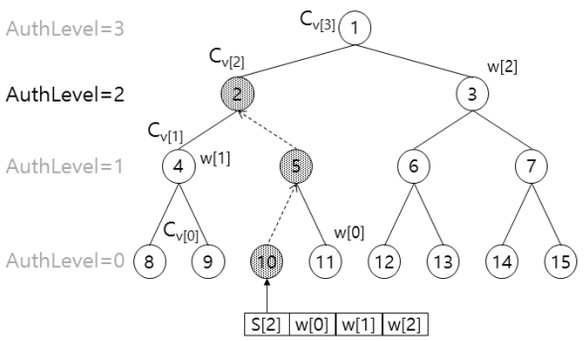


Fig. 2. MHT-based Content Authentication to Improve Computation Overheads

[13]에서는 인증 경로 상의 노드들의 노드 값 계산을 위한 해시 값 중복 계산 정도를 분석하고, 이러한 중복 요소로 인하여 발생되는 비효율성을 개선하기 위하여 계산된 해시 값을 저장한 후, 재사용하는 방법을 제안하였다. 이와 같은 중복 계산 요인을 개선함으로써 대용량 콘텐츠 전송 시 인증 소요 시간을 단축하였다.

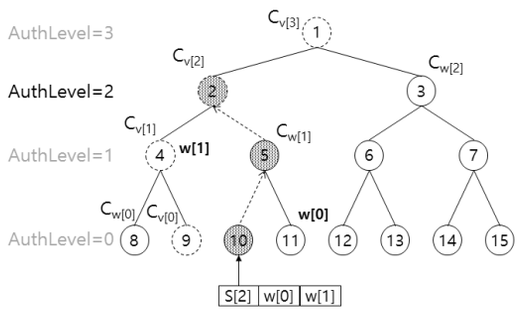
Fig. 2는 MHT의 해시 중복 계산을 개선한 기법을 간략하게 설명한다. 노드 $N[10]$에 할당된 세그먼트 $S[2]$를 인증하려 할 때, 기존 MHT 기법은 인증 경로 상의 모든 노드 $N[10], N[5], N[2], N[1]$의 노드 값들을 순차적으로 계산해야 한다. [13]의 개선안은 $N[9]$에 할당된 세그먼트 $S[1]$을 인증할 때, 검증된 노드 $N[9], N[4], N[2], N[1]$의 노드 값들을 인증 경로 캐시 $\{C_{v[i]} | i = 0,1,2,3\}$에 저장한 후, $S[2]$를 인증할 때 이 캐시 값들을 활용한다. $S[1]$과 $S[2]$의 인증 경로가 $N[2]$부터 동일하기 때문에, $S[2]$를 검증하기 위해 사용자는 $N[10]$부터 $N[2]$까지만 노드 값을 계산한 후, 인증 경로 캐시 $C_{v[2]}$에 저장된 $N[2]$의 노드 값과 계산된 $N[2]$의 값을 비교한다. $S[2]$의 인증이 완료되면, 계산 된 $N[10]$과 $N[5]$의 노드 값을 인증 경로 캐시에 저장한다. 비슷한 방법으로 $S[3]$을 인증하기 위해서 사용자는 $N[11]$과 $N[5]$의 노드 값만을 계산하면 충분하다.

3.3 MHT의 인증 정보 중복전송으로 인한 비효율성 개선

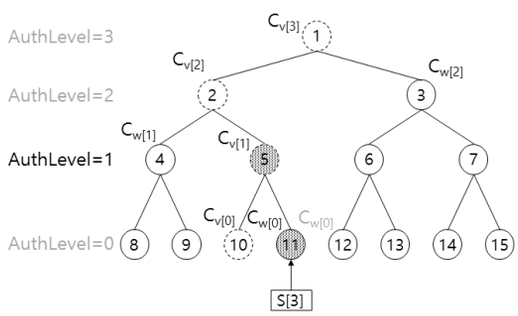
[14]에서는 MHT를 사용할 때 인증 정보가 중복되어 전송/처리됨을 지적하고, 이를 개선할 수 있는 방안이 제안되었다. 이를 위하여 [13]에서 제안한 단축된 인증 경로를 사용하는 동시에, 인증 정보의 중복 전송을 개선하기 위하여 동적 프로그래밍 기법을 활용한다.

Fig. 3은 MHT의 인증정보 중복 전송을 개선하기 위한 기법을 간략하게 설명한다. $S[2]$의 경우, 단축된 인증 경로의 계층 값 (Authentication Level)이 2이므로, $N[2]$의 노드 값까지만 계산하면 된다. 그러므로 $S[2]$ 전송 시, 전체 인증 정보 $\{w[0], w[1], w[2]\}$ 중에서 $w[0]$ (= $N[11]$)과 $w[1]$(=$N[4]$)만 전송되면 된다. 이렇게 전송된 인증 정보는 세그먼트 인증 후, 인증 정보 캐시 $\{C_{w[i]} | i = 0,1,2\}$에 저장된다.

특히, Fig. 3-b에서와 같이 홀수 순번의 세그먼트의 경우,



(a) Even Segment Verification



(b) Odd Segment Verification

Fig. 3. MHT-based Content Authentication to Improve Transmission Overheads

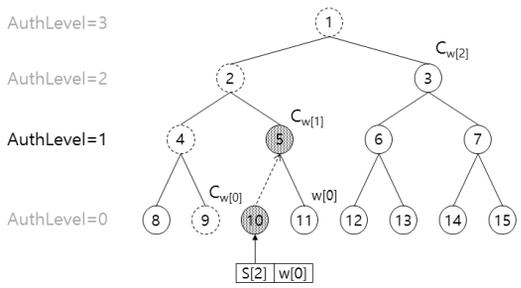
인증 정보할 필요가 없다. 예를 들어, $S[3]$ 인증을 위하여 사용자는 $S[2]$ 인증 후, $S[2]$의 해시 값 (= V_{10})을 $C_{w[0]}$에 저장한다. 즉, $S[2]$ 인증 완료 후, $C_{w[0]}$에는 V_{11} 값이 아닌 V_{10}값이 저장된다. 이와 같이 홀수 순번 $(2k+1)$ 세그먼트의 인증 경로의 계층 값은 항상 1이므로, 세그먼트 인증에 필요한 인증 정보는 $w[0]$ (= V_{2k}) 뿐이므로, 짝수 순번(2k) 세그먼트 인증 시 저장한 $C_{w[0]}$을 사용한다.

4. 인증 경로 단축 최적화

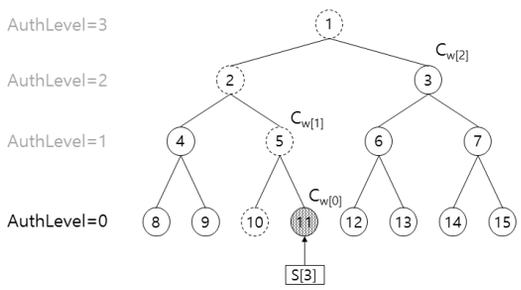
[14]에서 제안된 개선안은 인증 정보의 중복 전송 개선을 주요 해결 과제로 선정하여, 각 노드의 노드 값들이 인증 정보로 한 번 이상 전송되지 않도록 설계되었다. 특히 홀수 순번의 세그먼트 인증에 필요한 인증 정보 $w[0]$는 전송되지 않도록 하기 위하여, 인증 정보 캐시 $C_{w[0]}$를 다른 캐시와는 다르게 운영하도록 설계되었다.

본 절에서는 [14]에서 제안된 인증 정보 운영 방안을 이용하여 보다 단축된 인증 경로를 결정하는 기법을 제안한다. 인증 경로의 단축은 해시 값 계산 및 인증 정보 전송 효율성을 향상시킬 수 있다.

Fig. 4는 개선된 단축 인증 경로 결정 방법에 대한 설명이다. Fig. 4-a는 세그먼트 $S[1]$의 인증이 완료된 후, 인증 경로와 인증 정보 캐시 $\{C_{w[i]}\}$의 상태를 나타낸다. 이 때, 인증 정보 캐시에 저장되어 있는 값들은 사용자에게 이미 인증 받은



(a) Even Segment Verification



(b) Odd Segment Verification

Fig. 4. Proposed MHT-based Content Authentication

CalculationAuthenticationPathLevel:	
In	*Segment Number (sn),* *Binary Tree Depth (n)*
Out	*Authentication Level (level)*

```
SET mask := 1;
IF sn is 0, then OUPUT n ;
ELSE
   FOR level := 0 to n−1 :
      IF (i & mask) is not zero THEN
         OUTPUT level ;
      ELSE
         mask := mask << 1 ;
      END_IF
   END_FOR
OUTPUT level;
```

Fig. 5 Authentication Level Calculation

유효한 값이다. 즉, Fig. 4-a에서 인증 정보 캐시 $C_{w[0]}$, $C_{w[1]}$, 그리고 $C_{w[2]}$에 저장되어 있는 노드 값 V_9, V_5, 그리고 V_3은 모두 세스먼트 $S[1]$ 인증 시 사용자로부터 검증된 값이다. 그러므로 사용자는 세그먼트 $S[2]$를 인증하기 위하여 노드 $N[10]$부터 $N[1]$까지의 인증 경로 상의 모든 노드들의 노드 값들을 계산할 필요 없이, 노드 $N[5]$까지만 계산 후 인증하면 된다. 이 경우, [13]과 [14]의 제안보다 인증 경로의 계층 값을 줄일 수 있다. 또한, 각각의 세그먼트 인증 때마다 인증 경로의 최상위 계층 값은 이전 세그먼트 인증 시 검증된 인증 정보를 사용하므로 [13]에서와 같이 인증 경로 캐시를 별도로 운영할 필요가 없다.

PackagingData:	
In	*Segment Number (sn),* *sn-th segment (S[sn]),* *Witness Set for S[sn] (witness[]),* *sign*
Out	*data[sn]*

```
data[sn].segment :=S[sn] ;
level := CaclulationAuthenticationPahtLevel(sn) ;   // 1
IF sn is an even number, THEN                       // 2
    FOR k:=0 to level-1
        data[sn].witness[k]=witness[k]
    END_FOR
END_IF
IF sn is equal to 0, THEN
    data[sn].sign = sign                            // 3
END_IF
```

Fig. 6. Data Generation Pseudo Code

또한, Fig. 4-b와 같이 홀수 순번 세그먼트의 경우, 전송된 세그먼트의 해시 값을 계산한 후 $C_{w[0]}$에 저장된 값과 비교하여 세그먼트를 인증할 수 있기 때문에 별도의 인증 정보 전송이 요구되지 않는다.

Fig. 5는 각각의 세그먼트들의 인증 경로 계층 값을 결정하는 유사 코드(pseudo code)이다. 첫 번째 세그먼트 $S[0]$의 인증 경로 계층 값은 이진 트리의 깊이와 같다. 또한, 홀수 순번의 세그먼트의 인증 경로 계층 값은 모두 0이다. 이 외의 세그먼트의 인증 경로 계층 값은 이전 세그먼트의 인증에서 사용된 인증 정보와 해당 세그먼트 인증 경로의 전체 경로 중 최초 공통 노드의 계층 값으로 결정된다.

Fig. 6은 세그먼트 생성 시, 인증 정보를 전송할 데이터 패킷에 추가하는 유사 코드이다.

(1) 각각의 세그먼트에 대하여 세그먼트 번호에 따른 인증 경로 계층 값이 계산한다.

(2) 세그먼트 인증을 위해 인증 경로에 포함된 노드들 중에서 해당 인증 경로 계층 값 보다 작거나 같은 노드들의 노드 값만 계산하면 된다. 그러므로 인증 경로 계층 값 만큼의 인증 정보 전송이 필요하다. 인증 시 요구되는 인증 정보를 data에 패키징한다.

(3) 첫 번째 세그먼트의 인증은 전자서명 검증을 요구하므로 data에 서명값을 패키징한다.

Fig. 7은 수신된 세그먼트 인증을 위한 유사 코드이다.

(1) 세그먼트의 일련번호를 사용하여 Fig. 5에서 설명한 것처럼 해당 세그먼트의 인증 경로 계층 값을 계산한다. 세그먼트의 일련번호가 0이면, 해당 세그먼트에 할당된 리프 노드부터 루트 노드까지의 인증 경로 상의 모든 노드들의 노드 값들을 계산해야 되며, 그 이외의 경우는 인증 경로 계층 값에 따라 실제 인증 경로의 길이가 정해진다.

(2) 세그먼트의 해쉬 값을 계산한다.

(3) 1단계에서 계산된 인증 경로 계층 값을 이용하여, 최하

AuthenticationSegment:	
In	*sn,* *data[sn],* *witnessCache[]=(* $C_{w[]}$ *)*
Out	*Authentication Result,* *witnessCache[]*

```
level := CaclulationAuthenticationPahtLevel(sn) ; // 1
COMPUTE h=H( data[sn].segment ) ;          // 2
IF level is lager than 0
    FOR k := 0 to level-1:                 // 3
        COMPUTE h=H( data[sn].witness[k], h ) ;
    END_FOR
END_IF
IF sn is 0 THEN
    VERIFY data[sn].sign WITH h;           // 4
ELSE
    COMPARE C_w[level] WITH h;             // 5
END_IF
IF not verified, OUTPUT the result         // 6
IF it is verified and sn is an even number THEN
    FOR k:=0 to level-1 :
        C_w[k] := data[sn].witness[k] ;    // 7
    END_FOR
END_IF
OUTPUT the result of this verification/comparison;
```

Fig. 7. Segments Authentication Pseudo Code

위 노드부터 계산된 인증 경로 계층 값에 대응되는 상위 노드까지의 모든 노드 값들을 반복해서 계산한다. 이 때, 필요한 인증 정보는 데이터에 함께 패키징된 정보로부터 추출된 입력 값($data[sn].witness[]$)을 사용한다.

(4) 세그먼트의 일련번호가 0인 경우, 루트 노드의 노드 값을 계산한 후, 이를 이용하여 데이터에 패키징된 전자서명 값을 검증한다.

(5) 세그먼트의 일련번호가 0보다 큰 경우, 3단계에서 최종 계산된 인증 경로의 최상위 노드 값과 저장된 캐시 중 인증 경로 계층 값에 대응하는 캐시 값을 비교한다. 두 값이 같으면, 해당 세그먼트는 인증된 것으로 간주한다.

(6) 세그먼트 인증이 실패하면, 해당 결과를 통보한다.

(7) 세그먼트가 인증되면, 세그먼트 일련번호가 짝수인 경우에 한하여 전송된 인증 정보를 캐시에 저장한 후, 인증 결과를 통보한다.

Fig. 7의 유사 코드를 16 (=2^4)개의 세그먼트들로 구성된 콘텐츠 인증에 적용할 경우, Table 1과 같이 운영 된다: <sn>은 세그먼터의 일련번호를 의미한다. <level>은 해당 세그먼트를 인증하기 위해 MHT에서의 인증 경로의 길이를 의미한다. 즉, sn=0인 경우, 세그먼트 인증을 위해서 N[16]부터 N[1]까지 길이 4의 인증 경로 위의 모든 노드들의 노드 값들을 계산해야 된다. <AN>은 단축된 인증 경로의 최상위 노드를 의미한다. 즉, sn=0, 1, 2인 경우 각각의 인증 경로에서 $N[1], N[17], N[9]$까지의 노드 값을 각각 계산해야 된다는 의미이다. <Writing Witness Cache>는 해당 세그먼트 인증 후, 캐시에 저장된 노드 값들을 의미한다. 특히, 굵은 글씨로 기록된 정보는 해당 세그먼트 인증 후 새롭게 입력된 값을 의미하고, 괄호로 표시된 정보는 이전 세그먼트의 정보가 계속 유지됨을 의미한다.

Table 1. An Example of Segment Authentication

sn	level	AN	Writing Witness Cache			
			w[0]	w[1]	w[2]	w[3]
0	4	1	**17**	**9**	**5**	**3**
1	0	17	(17)	(9)	(5)	(3)
2	1	9	**19**	(9)	(5)	(3)
3	0	19	(19)	(9)	(5)	(3)
4	2	5	**21**	**11**	(5)	(3)
5	0	21	(21)	(11)	(5)	(3)
6	1	11	**23**	(11)	(5)	(3)
7	0	23	(23)	(11)	(5)	(3)
8	3	3	**25**	**13**	**7**	(3)
9	0	25	(25)	(13)	(7)	(3)
10	1	13	**27**	(13)	(7)	(3)
11	0	27	(27)	(13)	(7)	(3)
12	2	7	**29**	**15**	(7)	(3)
13	0	29	(29)	(15)	(7)	(3)
14	1	15	**31**	(15)	(7)	(3)
15	0	31	(31)	(15)	(7)	(3)

5. 성능 분석

세그먼트의 수를 $N=2^n$개라고 가정하자. 해시 계산량, 인증 정보 전송량, 그리고 캐시 저장 용량을 기반으로 Computation Overheads, Transmission Overhead, 그리고 Storage Overheads를 평가한다. Table 2는 2^n개의 세그먼트로 구성된 데이터를 인증할 때 MHT, [13], [14], 그리고 본 논문에서 제안한 개선안의 성능을 비교 분석한 결과이다.

개선안의 경우, 전송되는 인증 정보는 이진 트리의 각 노드의 오른쪽 자식 노드의 노드 값들이 한 번씩 전송된다. 즉, 2^n-1개의 인증 정보 전송만으로 충분하다. 이 경우, 전송되는 인증 정보의 수는 n의 크기에 따라 각각의 기술별로 전송량이 크게 차이가 난다. 예를 들어 $n=4$인 경우, MHT에 비해, 73%의 전송 효율 개선 효과가 발생한다. 그러나 $n=16$의 경우, 그 개선 효과는 94%까지 증가한다. 또한, 개선안의 전송 효율성 개선 정도를 [13], [14]과 비교할 때에도 각각 50%와 33%의 성능 개선 효과를 얻을 수 있다.

또한, 전송되는 인증 정보의 수 만큼의 해시 값 계산이 필요하다. SHA-512 기준으로 각각의 세그먼트의 크기를 4K

Table 2. Performance Evaluation

	Transmission Overheads	Computation Overheads	Storage Overheads
MHT	$n \times 2^n$	$n \times 2^n$	0
[13]	$2^{n+1}-2$	$2^{n+1}-2$ $(2^n+2^{n-1}-2)$	0 (n)
[14]	$2^n+2^{n-1}-2$	$3 \times 2^n-2$	$2 \times n$
개선안	2^n-1	2^n-1	n

바이트라 가정할 때, $n=16$인 경우 MHT는 105초의 기간이 소요되는 반면에 개선안의 경우, 7초가 소요되므로 전체 서비스 이용 지연을 개선할 수 있다.

이와 같은 성능 개선을 위하여 개선안은 전송된 인증 정보를 캐시에 임시 저장한 후, 재사용한다. 이를 위하여 2^n개의 세그먼트로 구성된 데이터 인증 시, n개의 캐시 저정 공간이 요구된다. 그러나 [14]의 경우, $2n$개의 캐시 저장 공간이 필요하기 때문에 본 논문의 개선안은 캐시 저장 공간의 효율성도 50% 개선되었다.

6. 결 론

CCN은 네트워크 노드에 캐싱된 데이터를 활용하여 네트워크의 효율성을 높이기 위해 제안되었다. 이와 같은 목적을 달성하기 위하여 CCN은 콘텐츠 생성자에게 집중되는 요청 메시지를 효과적으로 분산 처리할 수 있게 네트워크 노드에 캐싱 기능을 구현하고, 캐싱되어 있는 콘텐츠에 대한 요청 메시지를 수신한 네트워크 노드가 콘텐츠 생성자를 대신하여 해당 요청 메시지에 응답할 수 있게 설계되었다. 그러나 중간 노드에 의한 응답 처리는 사용자가 실제 콘텐츠를 전송한 노드를 식별할 수 없기 때문에 호스트 인증 기반의 보안 체계를 적용할 수 없다는 문제점과 함께 데이터의 위/변조가 가능하다는 취약점을 갖고 있다. 이러한 문제를 해결하기 위해 CCN은 MHT를 사용한 콘텐츠 인증 기법을 제안하고 있다. 그러나 MHT를 이용한 콘텐츠 인증은 해시 값을 중복해서 계산해야하고, 루트 노드의 해시 값 계산에 필요한 인증 정보를 중복으로 전송해야만 한다.

본 논문에서는 이러한 문제점을 개선하기 위하여 동적 알고리즘 기법을 적용하여 해시 값의 중복 계산 및 전송 문제를 개선함으로써 CCN의 인증 비효율성을 개선하였다. 본 논문에서 제안된 기법을 적용할 경우, 대용량 콘텐츠의 인증 정보 인증 시간을 60~85%까지 개선하는 동시에 전송량을 92%까지 개선할 수 있다.

References

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015 - 2020," *Cisco Public*, Feb. 3, 2016.

[2] "Cisco Visual Networking Index: Forecast and Methodology, 2015 - 2020," *Cisco Public*, Feb. 3, 2016.

[3] A. K. Pathan and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks," Tech Report, Univ. of Melbourne, 2007.

[4] E. Meshkova, J. Riihijarvi, M. Petrova, and P. Mahonen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks," *Computer Networks J.*, Vol. 52, No.11, pp.2097-2128, 2008.

[5] D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *ACM Sigcomm Comp. Comm. Review*, Vol.18, No.1, pp.106-114, Aug. 1988.

[6] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlmann, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, Vol.50, No.7, pp.26-36, Jul. 2012.

[7] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs and R. Braynard, "Networking Named Content," *5th International Conference on Emerging Networking Experiments and Technologies*, pp.1-12, 2009.

[8] D. Kim, "Content Centric Networking Naming Scheme for Efficient Data Sharing," *Journal of Korea Multimedia Society*, Vol.15, No.9, pp.1126-1132, 2012.

[9] D. Kim, "Trend and Improvement for Privacy Protection of Future Internet," *Journal of Digital Convergence*, Vol.14, No. 6, pp.405-413, Jun. 2016.

[10] D. Kim, "A Comparison Study on Data Caching Policies of CCN," *Journal of Digital Convergence*, Vol.15, No.1, pp. 327-334, Feb. 2017.

[11] R. Merkle, "Protocol for public key cryptosystems," *IEEE Sympo. Research in Security and Privacy*, Apr. 1980.

[12] B. Georg, "Merkle Signature Schemes, Merkle Trees and Their Cryptanalysis," Ruhr-Universität Bochum. Retrieved 2013-11-20.

[13] D. Y. Kim, "Improvement of the Data Authentication of CCN," *Journal of Digital Convergence*, Vol.15, No.8, pp. 341-349, Aug. 2017.

[14] D. Kim, "Network Overhead Improvement for MHT-based Content Authentication Scheme," *Journal of Digital Convergence*, Vol.15, No.8, pp.341-349, Jan. 2018.

김 대 엽

https://orcid.org/0000-0002-3100-6873

e-mail : daeyoub69@suwon.ac.kr

1997년 고려대학교 수학과(석사)

2000년 고려대학교 수학과(박사)

2000년~2002년 삼성 시큐아이 PKI실 차장

2002년~2012년 삼성전자 종합기술원 전문연구원

2012년~현 재 수원대학교 정보보호학과 조교수

관심분야 : 미래 인터넷, 블록체인, 콘텐츠 보안, M2M