Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.6, Dec. 2016 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2016.26.6.1375

컨텐츠 중심 네트워크에서 해시 체인 기반의 효율적인 데이터 인증 기법

서 석 충†‡
국가보안기술연구소

An Efficient Data Authentication Scheme for Content Centric Networking

Seog Chung Seo†‡
The Affiliated Institute of ETRI

요 약

본 논문에서는 CCN(Content Centric Networking)에서 세그먼트 데이터 인증에 대한 연산 부하와 전송량 부하를 크게 줄일 수 있는 해시 체인 기반의 데이터 인증 기법을 제안한다. 제안방법은 컨텐츠의 실제 정보를 담은 데이터 세그먼트들에 해시 체인 인증을 적용하고, 각 세그먼트들의 해시값들로 구성된 메타 부분에 대하여 해시 체인과 MHT(Merkle Hash Tree)를 적용한 2계층 인증 방법이다. 해시 체인과 MHT를 적절히 사용하여 해시 체인 방법의 효율성을 활용할 수 있으며, 또한, 해시 체인의 단점인 데이터 손실과 순서대로가 아닌(out-of-order) 전송 문제를 해결할 수 있다. CCNx 라이브러리에 구현하여 100Mbyte 전송 시, 연산 부하와 전송량 부하를 측정한 결과, 약, 2.596%와 1.803%만을 발생시키며, 이는 CCNx 라이브러리에 기본 탑재된 패킷 별 서명(per-packet signing), MHT 기반 서명과 비교하여 크게 개선된 것이다.

ABSTRACT

This paper presents an efficient hash chain-based data authentication mechanism which can considerably reduce the overhead of processing and transmission for authenticating segments in CCN. The proposed method makes use of hash chain and MHT(Merkle Hash Tree). At first, it applies hash chain methods for data segments and encodes them to Data part. Then, it constitutes Meta part with the hash values generated at the previous step and properly applies both hash chain method and MHT-based signing for not only achieving efficiency, but also mitigating the drawback(data-loss, out-of-order transmission) of hash chain method. We have implemented our method in the CCNx library and measured the performance. When transmitting 100Mbyte of content, the proposed method generates only 2.596% of processing overhead and 1.803% of transmission overhead.
Keywords: Content Centric Networking, Hash chain, Merkle Hash Tree, Data authentication

I. 서 론

스마트폰을 필두로 한 개인 모바일 컴퓨팅 환경의 도입과 가격대비 저장매체 용량의 빠른 발전으로 인하여 인터넷 트래픽이 기하급수적으로 증가하고 있다. 또한, 향후, IoT 환경에서는 데이터 폭증 현상이 더욱 가속화 될 것으로 전망되고 있다. 데이터 폭증은 네트워크 병목현상을 초래하고, 네트워크를 DDoS와 같은 네트워크 공격에 더욱 취약하게 만든다.

현재 인터넷에서 데이터 트래픽 폭증현상을 해결하기 위해 사용되는 방법으로서 CDN(Content Delivery Network)이 있다[1]. Akamai를 비롯한 CDN 기업들은 자체적으로 구축한 서버, 스토리

Received(08. 30, 2016), Modified(10. 20. 2016), Accepted(10. 20. 2016)
† 주저자, gegehe@gmail.com
‡ 교신저자, gegehe@gmail.com(Corresponding author)

지, 네트워크 인프라를 이용한 컨텐츠의 출판, 동기, 전달, 캐싱, 로드 밸런싱 등의 소프트웨어 솔루션을 이용하여 동영상, 게임 및 교육 컨텐츠 분야에서 트래픽을 분산시키고 있다.

V. Jacobson 외가 제안한 컨텐츠 중심 네트워킹(Content Centric Networking, CCN) 기술[2]은 현재 IP 기반의 네트워크를 근본 구조에서부터 바꾸는 것으로서, IP 주소 기반의 라우팅이 아닌 컨텐츠 이름 기반의 라우팅을 이용하여 컨텐츠를 전달한다. 또한, 컨텐츠가 전달될 때 라우팅 패스에 위치한 라우터들의 데이터 저장소에 저장됨으로써, 이후 동일한 컨텐츠가 요청될 때 근접 노드들에게 빠르게 컨텐츠를 전달할 수 있다. 즉, 사용자는 컨텐츠의 이름으로 네트워크에 컨텐츠를 요청하면, 컨텐츠의 이름을 이용하여 컨텐츠를 찾아간다. 이때, 라우팅 패스에 속한 라우터의 저장소에서 컨텐츠가 발견되면, 즉시 사용자에게 컨텐츠를 전달한다. CCN과 유사한 정보 중심 네트워크로서, DONA[3], NetInf[4], PSIRP[5] 등이 있으나, 현재는 CCN에 대한 연구가 가장 활발하다.

CCN에서 컨텐츠는 세그먼트 단위로 구성되어, 네트워크 라우터들의 저장소에 분산 저장되어 사용자들에게 전달되기 때문에 세그먼트 단위의 데이터 인증이 필요하다. CCN에서는 데이터 인증을 위하여 기본 데이터 포맷에 전자서명과 서명정보(Signed Info)를 포함시키고 있다. 하지만, 세그먼트 당 전자서명을 생성/검증하는 것은 큰 연산부하를 요구하며, 또한, 세그먼트 마다 전자서명을 포함하여 전달하는 것은 전송량 측면에서도 큰 부하를 발생시킨다. 세그먼트 당 요구되는 전자서명의 연산부하를 해결하고자 PARC에서 개발한 CCNx 라이브러리[6, 7]에서는 종합적(aggregate) 서명 방법의 일종인 Merkle Hash Tree(MHT)[8, 9]를 이용한다. MHT는 $n(=2^k, k > 0)$개의 세그먼트 블록에 대하여 해시트리를 구성하여 트리의 루트 노드에 대해서만 전자서명을 생성한다. n개의 세그먼트에 대하여 전자서명을 생성하는 횟수를 1번으로 줄일 수 있기 때문에 연산부하를 크게 줄일 수 있다. 하지만, 세그먼트를 검증하기 위해서는 전자서명값 뿐만 아니라, 서명 검증 시에 해시트리의 루트노드를 계산할 수 있는 필수 정보인 증거(witness) 정보가 포함되어야 한다(증거 정보에는 k개의 해시값들이 포함된다).

본 논문에서는 CCN에서 서명 생성/검증의 연산부하와 전송량 부하를 크게 줄일 수 있는 해시 체인 기반의 데이터 인증 기법을 제안한다. 제안방법은 먼저 컨텐츠를 구성하는 데이터 세그먼트들에 대하여 해시체인을 구성하여 Data part(데이터 부분)를 구성한다. 다음으로 해시 체인 인증의 단점인 data loss와 out-of-order 전송 문제를 해결하기 위하여, 세그먼트들의 해시값들로 구성된 Meta part를 구성한 후, 해시 체인과 MHT를 적용한다. 컨텐츠 요청자는 Meta part의 세그먼트들을 먼저 전송받아 검증한다. Meta part의 세그먼트는 해시 체인 또는 MHT 기반으로 검증될 수 있다. 또한, Meta part의 세그먼트는 Data part의 세그먼트들을 해시체인으로 검증할 수 있는 해시값들로 구성되어 있기 때문에, 이후 Data part의 세그먼트들을 전달받아 해시 체인으로 빠르게 검증할 수 있다. Meta part 인증과 Data part 인증의 2계층 인증 메커니즘을 통하여, 해시 체인 인증의 단점인 data loss와 out-of-order 전송 문제를 해결하였다. 또한, CCNx 라이브러리를 수정하여 제안 방법의 실효성을 검증하였다.

II. 관련연구

2.1 컨텐츠 중심 네트워크(Content Centric Networks, CCN)

CCN은 V. Jacobson[2] 외가 2009년에 현재 인터넷의 데이터 폭증에 따른 가용성(availability) 문제, 보안 문제 등을 해결하고자 새롭게 제안한 네트워킹 기술이다. CCN에서 사용자는 특정 IP를 가진 서버 또는 노드에 컨텐츠를 요청하는 것이 아닌, 컨텐츠의 이름으로 요청한다. CCN에서는 컨텐츠들이 전송될 때, 라우터를 포함한 네트워크 노드들의 캐시에 분산되어 저장된다. 따라서, 사용자가 컨텐츠를 요청하면, 해당 컨텐츠의 이름에 해당하는 컨텐츠를 자신의 캐시(또는 저장소)에 보유하고 있는 네트워크 노드가, 위치에 상관없이 컨텐츠를 즉각 사용자에게 전달한다. CCN에서는 컨텐츠들이 네트워크 노드에 분산되어 저장되고, 컨텐츠를 저장하고 있는 위치가 아닌 컨텐츠의 이름을 이용하여 여러 위치로부터 컨텐츠를 가져올 수 있기 때문에 데이터 폭증에 따른 네트워크 병목현상을 해결할 수 있다.

[Fig. 1.]은 CCN에서의 패킷 타입을 보여준다. 컨텐츠의 요청자가 컨텐츠의 이름정보를 포함한 Interest 패킷을 네트워크에 브로드캐스트하면,

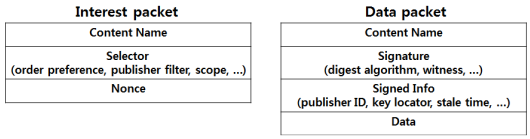

Fig. 1. CCN packet types

Interest 패킷의 컨텐츠 이름에 대응하는 컨텐츠를 보유한 노드는 Data 패킷을 요청자에게 전달한다. CCN에서는 컨텐츠의 데이터 세그먼트가 네트워크 노드들에 분산 저장되어 요청자들에게 전달되기 때문에, Data 패킷에 전자서명(Signature) 필드와 서명생성정보(Signed Info) 필드를 기본적으로 포함한다. 컨텐츠 수신자(요청자)는 Interest 패킷에 대응되는 Data 패킷을 수신한 후, Data 패킷의 Signed Info 필드의 정보를 이용하여 컨텐츠 생성자(제공자)의 공개키 정보를 얻어, Signature 필드의 유효성을 검증할 수 있다.

2.2 머클 해시 트리(Merkle Hash Tree, MHT) 기반 인증

컨텐츠를 구성하는 모든 Data 패킷에 대하여 전자서명을 생성/검증하는 것은 큰 연산부하를 가져오기 때문에, Data 패킷의 서명 생성/검증으로 인한 부하를 줄이기 위하여 PARC가 개발한 CCNx 라이브러리에서는 aggregate 서명의 일종인 MHT[8, 9] 방법을 적용하고 있다. MHT를 이용하여 2^k개의 데이터 세그먼트들에 대한 전자서명을 계산하기 위하여 먼저 세그먼트들에 대한 2진 해시 트리를 구성한 후, 트리의 루트 노드에 대하여 전자서명을 생성한다. 루트 노드에 대한 전자서명 값은 2^k개의 데이터 세그먼트에 대한 대표 전자서명 값이 된다. Data 패킷에는 서명 검증을 위하여 루트 노드에 대한 전자서명 값 뿐만 아니라, 해시 트리를 재구성할 수 있는 해시 패스 정보를 담은 증거(witness) 정보도 함께 전달해야 한다. [Fig. 2.]는 k=2인 경우에 대한, MHT 기반 서명 생성과정을 보이고 있다. m_1, m_2, m_3, m_4 세그먼트에 대하여 2진 해시 트리를 구성한 후, 루트 해시값인 $h_{1,2,3,4}$에 대하여 대표 전자 서명 값인 $Sign(h_{1,2,3,4})$를 계산한다. Data 패킷의 전자서명 필드에는 전자서명값 뿐만 아니라, 증거 정보도 함께 포함된다. 예를 들어, m_1 세그먼트의 경우에는 컨텐츠 수신자가 이진 해시 트리를 재구성할 수 있는

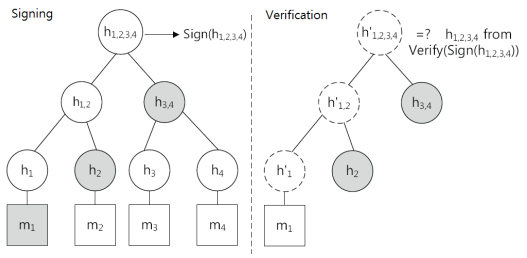

Fig. 2. MHT based signature generation and verification process($h_{i,j} = H(h_i||h_j)$, $h_i=H(m_i)$, where H=hash)

정보인 h_2와 $h_{3,4}$가 증거 정보가 된다. 따라서, m_1 세그먼트에 대한 Data 패킷은 m_1세그먼트에 대한 이름 필드, $Sign(h_{1,2,3,4})$와 $witness(h_2, h_{3,4})$로 구성된 Signature 필드, 생성자 ID와 공개키 위치 정보를 담은 Signed Info, 그리고 실제 데이터 부분으로 구성된다. m_1 세그먼트에 대한 데이터 패킷을 수신하였을 때, Fig. 2.의 오른쪽 그림과 같이 m_1에 대한 해시값인 h_1과 수신된 증거 정보에 포함된 h_2, $h_{3,4}$를 이용하여 루트 해시값인 $h'_{1,2,3,4}$를 계산할 수 있으며, $Sign(h_{1,2,3,4})$로부터 계산된 값인 $h_{1,2,3,4}$와 비교하여, 데이터 패킷의 무결성과 인증성을 검증할 수 있다.

[13]의 연구결과는 [11]에서 제안된 CVS(Code Verification Scheme)을 CCN 환경에 적합하도록 수정한 것으로서, 검증을 위해 저장해야하는 정보의 부하를 크게 줄였으나, 연산량 측면에서는 기존 CVS와 동일하다. [14]의 연구결과는 분산 네트워크 환경에서 효율적인 컨텐츠 인증방법을 제안하였으며, 이중 MHT를 적용하여 전송오류가 있는 환경에서 더욱 견고한 전송이 이루어질 수 있도록 하였다.

본 논문에서 제안하는 방법은 컨텐츠의 실제 정보를 담은 데이터 부분(Data part)에 해시 체인 인증을 적용하고, 각 세그먼트들의 해시값들로 구성된 메타 부분(Meta part)에 대하여 해시 체인과 MHT(Merkle Hash Tree)를 적용한 2계층 인증방법이다. 제안 방법의 데이터 부분에 적용된 인증방법은 [13, 14]의 연구결과와 유사하나, 해시값들로 구성된 메타 부분을 효율적으로 구성하고 이에 대하여 MHT와 해시 체인을 2중 적용함으로써, 기존 연구결과들보다 더욱 향상된 성능을 달성할 수 있다.

2.3 해시체인(Hash Chain) 인증

해시체인 인증 방법은 메시지의 해시값으로부터 원본 메시지를 복원하는 것은 계산적으로 불가능하다는 일방향 해시 함수의 특성을 이용한다[10, 11]. [Fig. 3.]은 n개의 패킷에 대하여 해시체인 인증을 적용한 예이다. 하나의 패킷은 실제 데이터 값과 다음 패킷에 대한 해시값을 포함한다. 즉, i번째 패킷 P_i는 $(Data_i||H_{i+1})$로 구성된다. 컨텐츠 생성자는 n개의 세그먼트로 구성된 컨텐츠에 대하여 n번째 패킷에서부터 시작하여, 연쇄적으로 1번째 패킷까지 해시 체인을 구성하고, 0번째 패킷에는 1번째 패킷의 해시값과 전자서명값을 계산하여 저장한다. 이때, 0번째 패킷에 포함된 전자서명은 해시체인 인증의 시작점이 된다. 컨텐츠 요청자는 0번째 패킷부터 시작하여 순차적으로 다음 패킷을 요청한다. 0번째 패킷을 수신하면, H_1에 대한 전자서명을 검증한다. 검증을 통과하면, H_1은 유효한 해시값이 되며, 이는 P_1 패킷을 수신하였을 때, P_1 패킷을 검증하는데 사용된다. 즉, 수신된 P_1 패킷에 대한 해시값을 계산하여, 이전에 검증된 H_1과 비교함으로써 검증을 수행한다. 해시체인이 n번째 패킷으로부터 역방향으로 연쇄적으로 구성되었기 때문에, i번째 패킷은 i+1번째 패킷을 검증할 수 있는 해시값을 포함한다.

컨텐츠 생성자는 n개의 패킷으로 구성된 컨텐츠의 인증을 위하여, n번의 해시연산과 1번의 전자서명 생성 연산을 수행하며, 컨텐츠 수신자는 n개의 패킷을 검증하기 위하여, 1번의 전자서명 검증 연산과 n번의 해시연산을 수행해야한다. 이는 패킷 당 전자서명 생성/검증 방법(per-packet-signing), MHT 기반(MHT-based signing) 인증 방법과 비교하여 연산측면에서 매우 효율적이다. 하지만, 해시체인 기반 인증의 경우, i번째 패킷이 전송 중에 소실될 경우, i번째 패킷이 재전송될 때까지, i+1번째 패킷 이후부터 패킷의 검증을 수행할 수 없다는 단점이 있다. 또한, 패킷이 순차적으로 전달이 되지 않을 경우(out-of-order 전송)에도 패킷에 대한 검증을 즉각적으로 수행할 수 없다. 따라서, 해시체인 기반 인증을 효율적으로 사용하기 위해서는 데이터 손실과 순서대로가 아닌 전송에 견고하도록 해야한다.

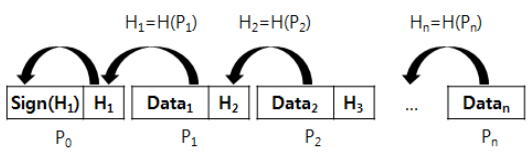

Fig. 3. Hash chain-based authentication(H=hash).

해시체인 기반인증 방법은 마지막 패킷에서부터 역방향으로 해시체인이 구성되기 때문에, 효율적인 해시체인 생성을 위하여 스트리밍 데이터보다는 VOD, 프로그램 설치 파일 등의 이미 생성되어있는 컨텐츠에 대하여 적용하는 것이 더욱 적합하다.

III. 제안방법

3.1 제안방법 개요

본 논문에서는 CCN에서 세그먼트 단위의 컨텐츠를 효율적으로 인증할 수 있는 방법을 제안한다. 제안 방법은 해시체인 인증 방법과 MHT 방법을 혼용하여, 각 방법의 단점을 보완한 것으로서, 데이터 전송량 측면과 연산 부하 측면에서, 기존 PARC 연구소의 CCNx 라이브러리에서 제공하는 세그먼트 인증방법보다 더욱 효율적이다.

[Table 1.]은 제안방법을 설명하기 위한 표기법 정리이다. 컨텐츠 C는 t-Kbyte 세그먼트 n개로 구성된다고 가정한다($|C|=(t\times n)$-Kbyte). 본 논문에서는 해시체인을 구성하기 위하여 sha1을 사용하였으나, 더 높은 안전성을 위하여 SHA-2 계열의 해시함수를 적용할 수도 있다.

제안방법의 컨텐츠 인증 데이터 생성은 두 단계로 구성된다. 첫 번째 단계에서 컨텐츠 C를 세그먼트로 나누어, 해시체인을 구성한다. 인코딩된 세그먼트 s_i^*는 데이터 세그먼트 s_i와 해시값 h_{i+w}로 구성된다 $(s_i^*=(s_i||h_{i+w}))$. s_i^*에 포함된 h_{i+w}는 s_{i+w}^* 세그먼트에 대한 해시값으로서, s_i^*가 수신되어 검증되면, 이후에 수신되는 s_{i+w}^*의 유효성을 검증할 수 있다. w는 세그먼트의 크기인 t-Kbyte를 해시값의 크기로 나눈 것이다. 예를 들어, 세그먼트의 크기가 4-Kbyte이고, 해시값의 크기가 16-byte라면 w는 256이 된다. 즉, s_i^*는 s_{i+256}^*의 유효성을 검증할 수 있는 해시값 h_{i+256}을 가지는 것이다. 이렇게 인코딩된 세그먼트들은 Data 파트를 구성한다.

두 번째 단계에서는 첫 번째 단계에서 생성된 해시값들을 이용하여 Meta 파트를 구성하는 것이다. 즉, Data 파트 구성 시에 생성된 해시값들로 세그먼트 S_i들을 구성한 후 이에 대하여 해시체인을 적용한다. Meta 파트에서 i번째 인코딩된 세그먼트 S_i^*

Table 1. Notations

C	Original content
|C|	size of C
|H|	size of hash
w	the number of hash values which can be stored in a segment of Meta part
C^*	Encoded content = (Meta part || Data part) = (S^* || s^*)
Hash	Hash operation(=sha1)
Sign	Signature generation
Verify	Signature verification
s	original segments
s_i	original i-th segment
s^*	encoded Data part
s_i^*	i-th encoded segment in Data part($s_i^*=(s_i\|\|h_{i+w})$)
|s|	Segment size
h	hash value in Data part
h_i	$h_i=Hash(s_i\|\|h_{i+w})=Hash(s_i^*)$
S_i	i-th segment in Meta data part($S_i=(h_{wi}\|\|\ldots\|\|h_{w(i+1)-1})$)
S_i^*	i-th encoded segment in Meta part($S_i^*=(h_{wi}\|\|\ldots\|\|h_{w(i+1)-1}\|\|H_{i+1})$)
S^*	encoded Meta part
H,	hash value in Meta data part
H_i	$H_i=Hash(S_i\|\|H_{i+1})$ where $i > 0$
H_i'	$H_i'=Hash(S_{i-1}^*)$ where $i > 0$
Sig_R, H_R	MHT에서 루트 서명값, MHT에서 루트 해시값

는 h_{wi}에서부터 $h_{w(i+1)-1}$까지의 해시값들과 i+1번째 세그먼트를 검증할 수 있는 해시값 H_{i+1}로 구성된다. 해시체인 계산이 모두 끝난 후에는, S_i^*에 대하여 MHT를 적용한다. 즉, 인코딩된 S_i^* 세그먼트들 각각에 대하여 해시값을 계산한 후, 이에 대하여 MHT를 적용한다. 위의 과정을 통하여 인코딩된 컨텐츠 C*는 Meta part와 data part로 구성되며, Meta part는 MHT가 적용되었기 때문에, Meta part의 세그먼트들은 대표 루트 서명값 Sig_R을 가지며, 또한 각 세그먼트별로 해시트리를 구성하는데 필요한 해시값들로 구성된 증거 정보를 포함한다.

컨텐츠를 구성하는 각 세그먼트를 검증하는 단계는 다음과 같다. 먼저 인코딩된 컨텐츠 C*에서 Meta part를 전송한다. Meta part는 Data part에 포함된 세그먼트들에 대한 해시값들을 세그먼트로 하여 해시체인과 MHT가 적용되었다. 따라서, Meta part의 세그먼트 하나가 검증되면, Data part의 w개의 세그먼트가 해시체인 방법을 이용하여 검증될 수 있다. Meta part의 세그먼트를 검증하기 위하여 해시체인 방법 또는 MHT 검증 방법을 적용할 수 있으며, 이는 세그먼트에 대한 해시값 보유 여부에 따라서 결정된다. 예를 들어, Meta part의 i번째 세그먼트를 전달받았을 때, i-1번째 세그먼트 검증의 결과로서, i번째 세그먼트에 대한 해시값을 보유하고 있다면 해시체인 검증방법을 적용하고, 그렇지 않다면 MHT 기반의 검증을 적용한다. 또한, i번째 세그먼트가 유효한 것으로 검증되면, 다음 i+1번째 세그먼트 검증을 위하여 i번째 세그먼트에 포함된 해시값을 기록해둔다. Meta part의 세그먼트는 Data part w개의 세그먼트들에 대한 w개의 해시값들로 구성되었기 때문에, Meta part의 i번째 세그먼트가 검증되면, Data part의 iw번째 세그먼트에서부터 {(i+1)w-1}번째까지의 세그먼트를 해시체인 방법을 적용하여 빠르게 검증할 수 있다. Meta part의 세그먼트와 Data part의 세그먼트를 검증하는 순서는, Meta part의 세그먼트를 모두 검증하고 Data part의 세그먼트를 검증하는 순차적 방법과 Meta part의 세그먼트를 검증하면서, Data part의 세그먼트를 검증하는 인터리빙 방법이 있다.

다음에 오는 절들에서는 제안하는 컨텐츠에 대한 인증데이터 생성방법과 생성된 인증데이터를 이용한 효율적인 컨텐츠 검증 방법에 대하여 자세히 다룬다.

3.2 인증 데이터 생성 방법

〔Alg. 1.〕은 3.1절의 제안방법 개요에서 기술한 컨텐츠 인증 데이터 생성 방법에 대한 알고리즘으로서, 과정 3에서 과정 10까지는 Data part의 세그먼트들에 대하여 해시체인을 적용하여 새롭게 인코딩된 Data part를 구성하는 단계이다. 알고리즘 설명의 편의성을 위하여 컨텐츠의 세그먼트 수는 w의 배수로 가정한다. 과정 3에서 과정 5는 Data part의 마지막 w개의 세그먼트(s_{n-1}부터 s_{n-w}세그먼트까지)에 대하여 해시를 적용하여, 이전 w개의 세그먼트들을(s_{n-w-1}부터 s_{n-2w}세그먼트까지) 인코딩하는 과정이다. 해시체인으로 인코딩된 세그먼트는 원본 세그먼

Algorithm 1. Authenticated Content Generation

INPUT: Content C, segment size |s|, private key PRK for signature generation
OUTPUT: Encoded content composed of meta data part and data part

1. C is divided into segments of |s|
2. n = |C|/|s| // #(segments) in Data part
3. **for i = n-1 to n-w do**
4. $h_i \leftarrow Hash(s_i)$
5. $s_{i-w}^* \leftarrow (s_{i-w}||h_i)$
6. **for i = n-2w-1 to 0 do**
7. $h_{i+w} \leftarrow Hash(s_{i+w}^*)$
8. $s_i^* \leftarrow (s_i||h_{i+w})$
9. **for i = w-1 to 0 do**
10. $h_i \leftarrow Hash(s_i^*)$
11. N ← ceil(n/w) // #(segments) in Meta part
12. **for i = N-2 to 0 do**
13. $H_{i+1} \leftarrow Hash(S_{i+1}^*)$
14. $S_i^* \leftarrow (S_i||\ H_{i+1})$
15. $H_0 \leftarrow Hash(S_0^*)$
16. **for i = N-1 to 0 do**
17. $H_{i+1}' \leftarrow Hash(S_i^*)$
18. $H_0' \leftarrow Hash(H_0)$
19. **MHTSign(H_0',...,H_N', PRK)**
20. **Return** (C^*).

트와 해시값을 포함한다($s_{i-w}^* \leftarrow (s_{i-w}||h_i)$). 과정 6에서 과정 8까지는 세그먼트 s_{n-2w-1}부터 세그먼트 s_0까지를 해시체인으로 인코딩한다. s_i번째 세그먼트를 인코딩할 때는 이미 인코딩된 s_{i+w}^*에 해시를 적용하여 얻은 h_{i+w}를 s_i와 연접한다($s_i^* \leftarrow (s_i||h_{i+w})$). 즉, s_i^* 세그먼트는 s_{i+w}^* 세그먼트를 검증할 수 있는 해시값을 포함한다. 따라서, 일단 s_i^* 세그먼트가 검증되면, s_{i+w}^* 세그먼트를 수신하였을 때, $Hash(s_{i+w}^*)$를 계산하여 s_i^*에 포함된 h_{i+w}와 비교함으로써 빠르게 검증할 수 있다. 과정 9에서부터 과정 10까지는 Data part의 세그먼트 s_0^*에서부터 s_{w-1}^*을 검증할 수 있는 해시값을 계산한다.

과정 11에서부터 과정 19은 과정 2에서부터 과정 10을 통해 계산된 해시값들로 세그먼트들을 구성한 후, 해시체인과 MHT를 적용하여 Meta part를 인코딩한다. Data part를 인코딩할 때 생성된 해시값의 수는 n개이고, 고정된 세그먼트 크기 안에 저장될 수 있는 해시값의 수가 w이기 때문에, Meta part의 세그먼트 수는 ceil(n/w)가 된다. 과정 12에서 과정 15를 거쳐 Meta part의 세그먼트들에 대하여 해시체인을 적용한다. 과정 16에서 과정 18은 해시체인으로 인코딩된 Meta part의 세그먼트들에 대하여 MHT 기반 서명을 적용하기 위하여

Algorithm 2. MHTSign

INPUT: N+1 Hash values to be signed, private key PRK for signing
OUTPUT: root signature, witness for each hash values

1. Find smallest L such that $N+1 \le 2^L$
2. Pad 2^L-(N+1) leaf with zero values
3. Construct binary hash tree with 2^L leaf nodes(N+1 hash values and 2^L-(N+1) zero padding values)
4. $Sig_R \leftarrow Sign_{PRK}(H_R)$
5. **Return**(binary hash tree, Sig_R)

해시 트리의 리프(leaf) 노드 해시값을 계산하는 단계이다. 과정 19는 〔Alg. 2.〕를 이용하여 리프 노드 해시값에 대하여 MHT 기반의 서명을 생성한다. 〔Alg. 2.〕는 N+1개의 해시값과 전자서명 생성을 위한 개인키를 입력받아, MHT 기반으로 전자 서명을 생성한다. 먼저, 과정 1에서부터 3을 수행하여, 입력된 해시값들을 리프 노드로 하여 이진 해시 트리를 구성한다. 과정 4에서는 이진 해시 트리의 루트 해시 노드에 대하여 전자서명을 생성한다.

Meta part의 세그먼트들은 위와 같이 해시체인 인증 방법과 MHT 기반 서명이 함께 적용되어 있기 때문에, 세그먼트를 검증할 수 있는 해시값을 가지고 있는 경우에는 해시체인 기반 검증 방법을 적용하고, 그렇지 않을 경우에는 MHT 기반 검증을 수행한다.

〔Fig. 4.〕는 100-Mbyte 컨텐츠에 대하여 제안 인증 데이터 생성 방법을 적용한 예제이다. 세그먼트의 크기는 4-Kbyte로 가정했기 때문에, Data part의 세그먼트 수는 25,600개이다. 해시 알고리즘은 SHA-1을 적용하였으나, 20-Byte 중, 16-Byte만을 해시값으로 사용한다. Data part의 세그먼트들에 해시체인을 적용한 후, 0번째부터 25,343번째까지의 인코딩된 세그먼트는 w번째 뒤에 있는 세그먼트를 검증할 수 있는 해시값을 포함한다. 마지막 w개의 세그먼트는 이후에 검증할 세그먼트가 없기 때문에, 해시값을 포함하지 않는다. Meta part의 세그먼트들은 Data part에서 생성된 해시값들로 구성된다. 즉, Meta part의 첫 번째 세그먼트는 h_0에서부터 h_{255}를 포함하며, 이는 Data part의 첫 번째 256개의 인코딩된 세그먼트를 검증할 수 있는 해시값들이다. Meta part의 세그먼트들도 마찬가지로 해시체인이 적용되었으며, 견고한 전송을 위하여 해시체인으로 인코딩된 세그먼트들에 대하여 MHT 기반의 전자서명을 적용한다.

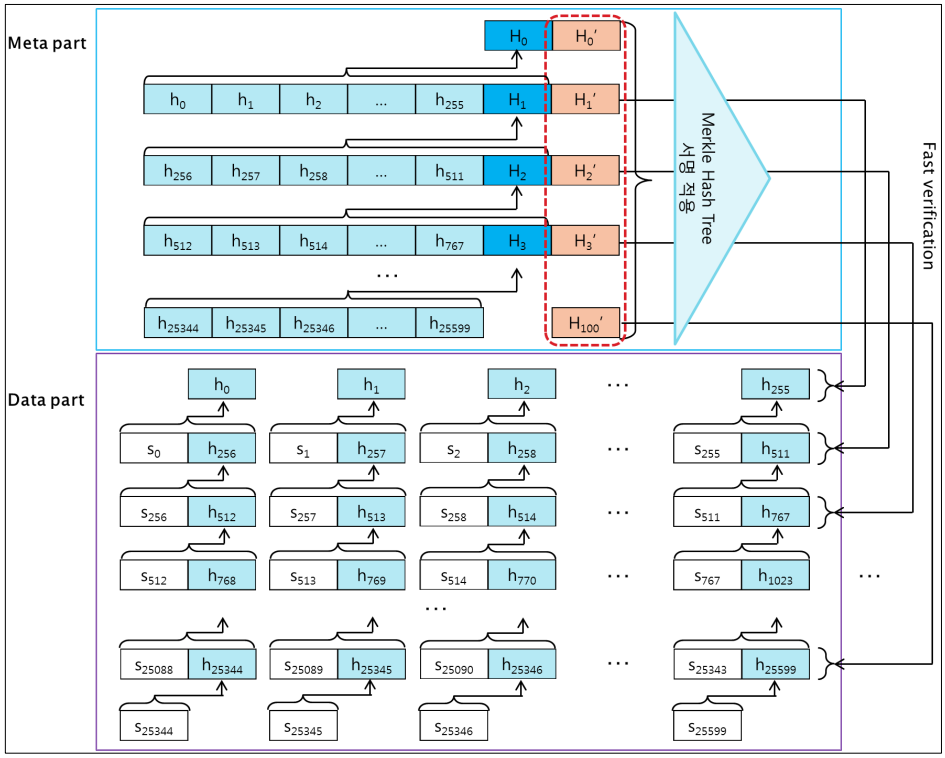

Fig.4. Proposed Content Structure(Assume that the size of content file is 100Mbytes and it is divided into 25,600 segments of 4Kbytes.)

3.3 컨텐츠 세그먼트 검증 방법

3.2절에서는 본 논문에서 제안하는 컨텐츠에 대한 효율적인 인증데이터 생성 방법에 대하여 살펴보았다. 본 절에서는 3.2절에서 제시된 방법을 통하여 생성된 인증데이터를 이용하여, 세그먼트들을 효율적으로 검증하는 방법을 다룬다.

컨텐츠가 효율적으로 검증될 수 있도록, 먼저, Meta part의 세그먼트를 전송한다. Meta part에 속한 하나의 세그먼트를 검증함으로써, Data part의 w개 세그먼트를 해시체인으로 빠르게 검증할 수 있다. 예를 들어, Fig.4.에서 Meta part의 첫 번째 세그먼트 S_0*는 해시값인 ($h_0||...||h_{255}$)로 구성되며, 이는 Data part의 s_0*에서부터 s_{255}*까지 세그먼트를 해시체인으로 검증할 수 있는 값들이다.

Meta part의 세그먼트는 MHT 기반 또는 해시체인 기반으로 검증될 수 있다. Meta part에 포함된 모든 세그먼트들은 MHT 기반으로 검증될 수 있기 때문에, Data loss와 out-of-order 전송에 견고하다. 또한, 해당 세그먼트를 검증할 수 있는 해시값이 이미 검증되어 저장되어 있다면, 해시체인 방법을 적용하여 빠르게 검증할 수 있다. 제안 방법에서는 이미 검증된 해시 체인값을 저장하기 위한 테이블인 AHT(Authenticate Hash Table)를 사용한다. 예를 들어, S_i* 세그먼트를 검증할 수 있는 해시값인 H_i값이 AHT에 이미 검증되어 들어있다면, 해시체인 검증을 적용하고, 그렇지 않다면 MHT 기반으로 세그먼트의 유효성을 검증한다. S_i* 세그먼트가 검증되면, S_i*에 포함된 H_{i+1} 해시값을 AHT에 저장하고, S_{i+1}* 수신 시, AHT에 저장된 H_{i+1}값을 이용하여 해시체인 기반 방법으로 검증한다.

〔Fig. 5.〕는 제안하는 세그먼트 검증방법에 대한 절차도이며, 그림의 윗부분은 Meta part의 세그먼트에 대한 검증 절차를, 아랫부분은 Data part의 세그먼트에 대한 검증 절차를 나타낸다. Meta part의 세그먼트를 수신하였을 때, 먼저, AHT에서 해당 세그먼트를 검증할 수 있는 해시값이 이미 검증되어 저장되어 있는지를 확인한다. 존재한다면, 검색된 해시값을 이용하여 해시체인 기반으로 해당 세그먼트를 검증하고, 존재하지 않는다면, MHT 기반 서명 검

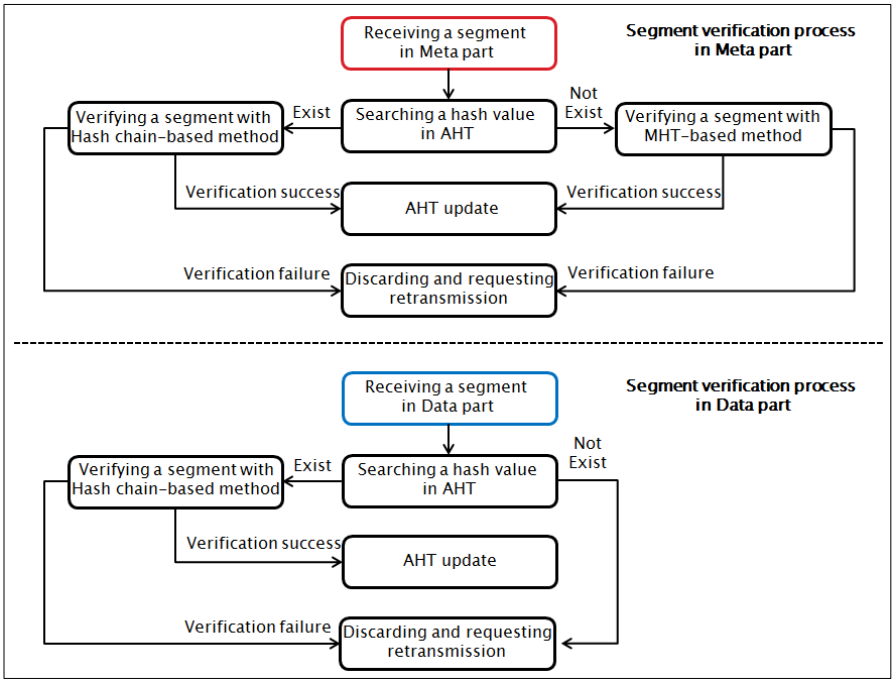

Fig.5. Proposed content segments verification process(Above: Meta part, Below: Data part)

증을 이용하여 세그먼트를 검증한다. 세그먼트가 유효한 것으로 검증되면, 세그먼트에 포함된 해시값을 다음 세그먼트 검증을 위하여 AHT에 저장하여 업데이트한다. 만약, 세그먼트가 유효하지 않은 것으로 판단되면 이를 폐기하고, 전송자 측에 재전송을 요청한다.

Data part의 세그먼트 전송은 Meta part의 모든 세그먼트 전송이 완료되어 검증된 후 수행되는 순차적(sequential) 방법과 Meta part에 속한 하나의 세그먼트가 검증되면, Data part에서 연관된 w개의 세그먼트를 전송하는 인터리빙(interleaving) 방법이 있다. Data part의 세그먼트 검증 시에도 마찬가지로, AHT에 해당 세그먼트를 검증할 수 있는 해시값이 존재하는지를 검색한 후, 존재할 경우에 해시체인 기반의 검증을 수행한다. 세그먼트가 유효한 것으로 검증되면, AHT를 업데이트하여, 다음 세그먼트 검증에 필요한 해시값을 저장한다. 순차적 방법이 적용될 경우, AHT에는 Data part의 모든 세그먼트를 검증할 수 있는 해시값들이 포함되어 있기 때문에, Data part의 세그먼트 검증 후, AHT를 업데이트할 필요가 없다. 하지만, 인터리빙 방법이 적용된 경우, AHT는 Data part의 세그먼트들에 대한 해시값을 부분적으로 가지고 있기 때문에, Data part의 세그먼트를 검증한 후, 해당 세그먼트에 포함된 해시값이 AHT에 없다면, AHT를 업데이트해야 한다.

IV. 성능 및 분석

4.1 Data loss, out-of-order에 대한 견고성

3장에서 CCN에서 세그먼트 단위의 컨텐츠 전송 시에 세그먼트를 효율적으로 인증할 수 있는 방법을 제안하였다. 제안방법에서 실제 데이터를 담은 Data part에 대하여 해시체인을 적용한 후, Data part에서 생성된 해시값들을 다시 세그먼트들로 구성하여 Meta part를 구성하였다. Meta part의 세그먼트들에는 해시체인을 적용한 후, Data loss와 out-of-order 전송에 견고하도록 하기 위하여 MHT 기반 서명을 적용하였다.

제안방법을 이용하여 인코딩된 컨텐츠는 먼저 Data part의 세그먼트들을 검증할 수 있는 해시값들로 구성된 Meta part의 세그먼트들부터 전송한다. Meta part의 세그먼트들은 해시체인 뿐만 아

니라 MHT 기반 서명이 적용되었기 때문에, 세그먼트가 전송 중에 누락되거나 순서가 뒤바뀌어 도착하더라도, 수신자 측에서 대기없이 즉각적으로 세그먼트를 검증할 수 있다. 즉, 수신자 측의 AHT에 세그먼트를 검증할 수 있는 해시값이 있을 경우 해시체인 기반으로 세그먼트를 검증하고, 그렇지 않을 경우에는 MHT 기반 서명 방법으로 생성된 루트 서명값과 증거 정보를 이용하여, 세그먼트를 독자적으로 검증한다. 따라서, 해시체인 기반 데이터 인증 방법의 단점인 Data loss와 out-of-order 전송 문제를 보완하였다. 또한, 세그먼트가 검증된 후에는 세그먼트에 포함된 해시값을 AHT에 반영하여 다음 세그먼트 검증 시에 사용될 수 있도록 한다. 예를 들어, Meta part의 세그먼트 S_i*는 Data part의 w개의 세그먼트(s_{wi}*에서부터 s_{2wi-1}*)를 해시체인으로 검증할 수 있는 해시값들을 포함하고 있기 때문에, 이를 AHT에 반영하여, Data part의 세그먼트들에 대하여 해시체인 검증을 즉각적으로 적용할 수 있도록 한다. Data part의 세그먼트 s_i* 역시 세그먼트 s_{i+w}*를 검증할 수 있는 해시값을 포함하고 있기 때문에, Meta part의 세그먼트 전송과 Data part 세그먼트의 전송을 인터리빙하게 수행할 수 있다. 결론적으로, Data part에 적용된 해시체인 기반 세그먼트 검증의 원활한 동작을, Meta part의 세그먼트들의 검증을 통하여 보장해주기 때문에, 제안방법은 Data loss와 out-of-order 전송에 견고하다.

4.2 제안 방법의 구현

제안방법을 CCNx 라이브러리를 수정하여 적용하였다. 제안방법을 구현하기 위하여 CCNx 0.7.2 버전[7]을 사용하였으며, CCNx에 포함된 C 라이브러리를 수정하여 제안방법을 구현하였다.

CCN에서 컨텐츠 전송은 컨텐츠 요청자가 컨텐츠 제공자에게 컨텐츠 이름으로 데이터를 요청하면서 시작된다. 즉, 컨텐츠 제공자가 자신이 제공할 컨텐츠에 대하여 컨텐츠의 이름으로 자신의 저장소에 등록한 후, 요청자가 컨텐츠 이름으로 네트워크에 컨텐츠를 요청하면, 컨텐츠 이름에 해당하는 컨텐츠를 보유한 생성자가 컨텐츠를 전송하기 시작한다. CCNx 라이브러리에서 하나의 컨텐츠는 control 데이터, Metafile, 그리고 실제 데이터를 담은 Datafile로 구성되며, 요청자는 control 데이터, Metafile, Datafile 순서로 컨텐츠를 요청한다. 이때, 컨텐츠는 세그먼트 단위로 전송되며, 컨텐츠 이름에 세그먼트 번호 정보를 붙여서 유일하게 식별한다. 요청자는 세그먼트 번호를 포함한 컨텐츠 이름에 대한 Interest 메시지를 전송하여 컨텐츠를 요청하며, 제공자로부터 해당 세그먼트가 수신되면 세그먼트를 검증한다. 세그먼트가 검증되면, 세그먼트 번호를 증가시켜서 다음 세그먼트를 요청하며, 실패할 경우, 해당 세그먼트에 대하여 재전송을 요청한다.

위에서 설명한 CCNx 라이브러리를 통한 컨텐츠 전송 절차에 제안방법을 구현하였다. 제안방법에서 컨텐츠 제공자는 요청자에게 제공하려는 컨텐츠에 대하여 오프라인(offline)에서 해시체인을 적용하여, Data part와 Meta part로 구성되도록 컨텐츠를 인코딩한다(제안방법은 인코딩된 Data part와 Meta part를 Datafile과 Metafile에 각각 저장한다.). 이때, 오프라인에서는 Meta part에 대하여 MHT 기반 서명을 적용하지 않는다[1]. CCNx 라이브러리에서 컨텐츠 요청자는 control 데이터, Metafile, Datafile 순서로 컨텐츠를 요청하기 때문에, 제안방법에서는 Metafile에 대한 전송 요청이 발생할 때, Meta part의 세그먼트들에 online으로 MHT 기반 서명을 적용하여, Meta part 세그먼트들에 대한 루트 서명을 계산한다. 루트 서명값과 해시트리가 계산되면, 순차적으로 Meta part의 각 세그먼트에 대하여, 루트 서명값, 그리고 해시트리를 구성할 수 있는 증거 정보를 포함하여 전송한다.

컨텐츠 요청자는 Metafile에 대한 세그먼트를 수신 하여, MHT 기반 또는 해시체인 기반으로 세그먼트를 검증한다. 이때, 각 세그먼트에 대한 검증이 성공하면, AHT를 업데이트하여, 이후 세그먼트들에 대하여 해시체인으로 검증할 수 있도록 한다. Metafile을 구성하는 모든 세그먼트에 대한 검증이 완료되면, Data part를 포함한 Datafile을 요청하며, Datafile에 포함된 세그먼트들에 대해서는 AHT에 포함된 해시값을 이용하여 해시체인으로 유효성을 검증한다.

1) 해시알고리즘의 특성 상, 동일한 데이터에 대한 해시값은 일치한다. 따라서, 컨텐츠에 대하여 해시체인을 적용하는 것은 오프라인에서 수행할 수 있으며, 이는 고정된 컨텐츠이다. 반면, 전자서명값은 사용되는 서명키, 패딩값에 따라서 달라지는 동적 데이터이기 때문에 오프라인에서 생성하지 않는다.

Table 2. Transmission overhead comparison(When transferring a 100MByte content. Null-Sig Mode does not equip any segment authentication methods.). The amount of Control data, Metafile, Datafile, and Total are expressed as byte.

	Null-Sig Mode	Proposed	per-packet -signing	MHT-based (size=2)	MHT-based (size=4)	MHT-based (size=8)	MHT-based (size=16)
Control data	410	1,008	1,008	1,008	1,008	1,008	1,008
Metafile	410	460,808	984	984	984	984	984
Datafile	109,363,116	110,873,825	117,017,815	118,656,215	119,526,615	120,397,015	121,267,415
Total	109,363,936	111,335,641	117,019,807	118,658,207	119,528,607	120,399,007	121,269,407
Overhead (%)	-	1.803	7.000	8.498	9.294	10.090	10.886

Table 3. Performance comparison(When transferring a 100MByte content. Null-Sig Mode does not equip any segment authentication methods.)

	Null-Sig Mode	Proposed	per-packet -signing	MHT-based (size=2)	MHT-based (size=4)	MHT-based (size=8)	MHT-based (size=16)
performance (Mbps)	141.74	138.06	40.79	75.23	127.36	129.19	129.58
Overhead (%)	-	2.596	71.222	46.924	10.145	8.854	8.579

4.3 성능비교

본 절에서는 제안방법의 성능을 PARC에서 개발된 CCNx 라이브러리에 구현된 패킷 별 서명방법, MHT 기반 서명방법과 성능, 전송량 측면에서 비교한다.

1Gbps 로컬 LAN 상에서 컨텐츠 제공자-허브-컨텐츠 요청자의 세 대의 Laptop(Intel i7 2.3Ghz 탑재)으로 구성된 토폴로지를 이용하여 실험하였다. Ubuntu 11.04 리눅스 상에서 CCNx 0.7.2 버전을 설치하여 제안방법을 구현하였으며, 컨텐츠 전송 실험을 위하여 ccnputfile(제공자 측)과 ccngetfile(요청자 측)을 수정하여 이용하였다.

ccnputfile, ccngetfile을 이용하여 100MByte 크기의 파일을 전송할 때, 제안방법의 전송량 부하와 연산량 부하를 패킷 별 서명방법, MHT 기반 서명방법과 비교하였다. 이때, 전자서명 알고리즘과 해시 알고리즘으로 RSA-PSS 1024-bit[12]와 SHA-1을 사용하였다.

세그먼트 검증에 대한 전송량과 연산량 부하의 베이스라인을 잡기 위하여, 세그먼트 인증 메커니즘을 배제한 컨텐츠 전송 속도를 측정하였다(Null-Sig mode). 그리고 MHT-based singing 방법은 트리 사이즈를 16까지만 증가시켰다[2].

[Table 2.]과 [Table 3.]는 제안방법과 패킷 별 서명방법, MHT 기반 서명방법을 각각 전송량 부하와 성능 부하측면에서 비교한다. Null-Sig Mode는 인증 메커니즘을 배제하고 컨텐츠를 전송한 결과이며, 이를 베이스라인으로 잡고, 제안방법과, 패킷 별 서명방법, MHT 기반 서명방법의 부하를 측정하였다. [Table 2.]은 세그먼트 인증 메커니즘으로 인한 전송량 부하를 비교하고 있다. 전송량 부하를 측정하기 위하여, 인증 메커니즘 별로, 컨텐츠 요청자가 수신한 컨텐츠의 크기를 기록하도록 하였다. 컨텐츠를 구성하는 Control data, Metafile, Datafile 각각에 대하여 Null-Sig Mode에서 수신되는 양과 비교하였으며, 이를 모두 더한 전송 총량을 이용하여 전송량 부하를 측정하였다.

패킷 별 서명방법은 Datafile의 모든 세그먼트에 대하여 전자서명을 생성하여 전송한다. 즉, 하나의

2) MHT-based signing 방법에서 해시 트리의 크기를 크게 할수록, 수행되는 전자서명 횟수가 줄어드나, CCNx 라이브러리에서 제공하는 MHT-based signing 방법의 성능을 측정한 결과, MHT의 트리 크기를 32 이상으로 증가할 경우에는 전송량 부하에 비하여, 성능 향상이 크지 않았다.

세그먼트 전송 시에, 세그먼트를 검증할 수 있는 서명값과 서명값을 검증 때 사용되는 공개키 정보인 Signed Info를 포함한다. MHT-based signing은 2^k(k=1,2,3,4)개의 세그먼트들에 대하여 해시트리를 구성하여, 루트 해시에 대하여 서명을 생성하여 전송한다. MHT-based signing은 각 세그먼트당 루트 서명값과 Signed Info 뿐만 아니라, 해시트리를 구성하는데 사용되는 증거 정보도 포함하기 때문에 해시트리의 크기가 클수록 Datafile에 대한 전송량이 늘어난다. 제안방법은 인코딩된 Meta part를 Metafile에 저장하기 때문에 Null-Sig Mode, 다른 인증방법들과 비교하여 사용하는 Metafile의 크기는 크지만, 전자서명과 Signed Info 정보 대신에 해시체인만을 포함하여 전송하기 때문에, Datafile의 크기가 패킷 별 서명방법과 MHT 기반 서명방법과 비교하여 매우 작다. 제안방법으로 인한 전송량 부하는 약 1.803%이며, 이는 CCNx 라이브러리에서 제공하는 per-packet-signing(7%), 그리고 MHT-based signing(8.498%~10.886%)과 비교하여 더욱 효율적이다.

〔Table 3.〕는 Null-Sig Mode 대비, 제안방법으로 인한 연산부하를 패킷 별 서명방법, MHT 기반 서명방법과 비교한다. Null-Sig Mode의 경우 전송속도는 141.74Mbps 였으며, 제안방법의 경우 138.06Mbps로 약 2.596%의 연산부하를 발생시켰다. 반면, 패킷 별 서명방법의 경우 모든 세그먼트에 대하여 전자서명을 생성하고, 검증하기 때문에 큰 연산부하를 발생시켜, 약 71.222%의 부하를 발생시킨다. MHT-based signing는 해시트리의 크기를 증가시킬 경우, 연산부하가 줄어드나, 크기 16부터는 연산부하가 크게 줄어들지 않는다.

V. 결 론

본 논문에서는 CCN에서 세그먼트 인증을 효율적으로 수행할 수 있는 해시체인 기반의 인증기법을 제안하였다. 제안방법은 2계층 인증기법으로서, 컨텐츠의 실제 데이터를 담은 Data part에 대하여 해시체인을 적용한 후, Data part에서 계산된 해시값들을 이용하여 다시 Meta part를 구성한다. Meta part에는 효율성을 위하여 해시체인 방법이 적용되었으며, 또한 해시체인의 단점인 데이터 손실과 순서대로가 아닌 전송을 보완하기 위하여 MHT 기반 서명이 적용하였다.

PARC에서 개발한 CCNx 라이브러리에 제안방법을 구현하였으며, 제안방법으로 인한 전송량 부하와 연산 부하는 각각 약 1.803%와 2.596%로서 세그먼트 인증으로 인한 부하를 크게 개선하였다. 또한, CCNx 라이브러리에서 기본적으로 제공하는 패킷 별 서명방법, MHT 기반 서명방법과 전송량 부하와 연산 부하 측면에서 비교하여 전송량 부하와 연산 부하가 크게 향상된 것을 확인하였다.

제안방법은 해시체인에 기반을 둔 것이기 때문에, 동적으로 생성되는 컨텐츠보다는 VOD, 프로그램 설치 파일 등 이미 생성되어있는 컨텐츠에 대하여 적용하는 것이 더욱 적합하다.

향후 연구로는 제안방법에 적용된 전자서명, 해시 알고리즘의 비도를 확장하여 성능을 특정할 것이며, 또한 프로토콜 측면에서의 제안방법에 대한 안전성 분석도 추가적으로 수행할 것이다. 제안방법에서 메타부분에 대한 전송은 빠르고 신뢰성있게 수행되어야 하기 때문에, 향후, 에러정정코드(Forward Error Correction〔15〕)에 대한 추가 연구도 수행할 예정이다.

References

〔1〕 A. Vakali, and G. Pallis, "Content Delivery Networks: Status and Trends," IEEE Internet Computing, Vol.7, No. 6, pp. 68-74, Nov. 2003.

〔2〕 V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," Proceedings of the 5th international conference on Emerging networking experiments and technologies(CoNEXT'09), pp. 1-12, Dec. 2009.

〔3〕 T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica, "A Data-oriented (and beyond) Network Architecture," Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications(SIGCOMM07), pp. 181-192, Aug. 2007.

〔4〕 NetInf project, "http://www.netinf.org"

〔5〕 PSIRP project, "http://www.psirp.org"

〔6〕 The Content Centric Networking(CCNx) Project, "http://www.ccnx.org"

〔7〕 CCNx Project, "http://github.com/ProjectCCNx/ccnx"
〔8〕 R. Merkle, "A Digital Signature Based on a Conventional Encryption Function," Proceedings on Advances in cryptology(CRYPTO'87), LNCS 293, pp. 369-378, Aug. 1987.
〔9〕 R. Merkle, "A Certified Digital Signature," Proceedings on Advances in cryptology(CRYPTO'89), LNCS 435, pp. 218-238, 1989.
〔10〕 J. Deng, R. Han, and S. Mishra, "Secure Code Distribution in Dynamically Programmable Wireless Sensor Networks," Proceedings of the 5th international conference on Information processing in sensor networks(IPSN'06), pp. 292-300, Apr. 2006.
〔11〕 Sangwon Hyun, Peng Ning and An Liu, "Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks," Proceedings of the Seventh International Conference on Information Processing in Sensor Networks (IPSN'08), pp. 445-456, May. 2008.
〔12〕 Digital Signature with Appendix - Part2: Integer Factorization Based Mechanisms, "ISO/IEC 14888-2:2008," 2008.
〔13〕 DaeYoub Kim, Jaesung Park, "Efficient Contents Verification Scheme for Contents-Centric-Networking", The Journal of Korea Information and Communications Society, Vol. 39B, No.04, pp. 234-241, Apr. 2014.
〔14〕 DaeYoub Kim, "A Efficient Contents Verification Scheme for Distributed Networking/Data Store", Journal of the Korea Institute of Information Security & Cryptology, Vol. 25, No. 4, pp. 839-847, Aug. 2015.
〔15〕 Luigi Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", ACM SIGCOMM Computer Communication Review, Vol. 27, No. 2, pp. 24-36, Apr. 1997.

〈저자소개〉

사 진

서 석 충 (Seog Chung Seo) 정회원
2011년 8월: 고려대학교 정보보호학과 박사
2013년 11월: 삼성전자 종합기술원 전문연구원
2014년 4월: 삼성전자 DMC 연구소 책임연구원
2014년 5월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 미래인터넷 보안, CMVP, 공개키 암호, 최적화 구현, 병렬처리