제1회 한국사이버테러정보전학회 춘계학술발표대회 (2004.5)

CDN 기반의 디지털콘텐트 전송에서 안전한 그룹별 관리 기법 설계

고일석*, 나윤지**
*충북과학대학 전자상거래과, **충북대학교 컴퓨터공학과

A Design of Secure Group-Management Method on the Digital Content Delivery based on CDN

Il Seok Ko*, Yun Ji Na**
*Dept. of e-commerce, Chungbuk Provincial University
**Dept. of Computer Engineering, Chungbuk University

요 약

중앙집중 구조의 멀티미디어 콘텐트 서비스에서는 서버의 과부하 문제와 네트워크 트래픽의 급격한 증가 문제가 발생한다. 최근에는 이러한 문제점을 해결하기 위한 디지털콘텐트 전송기술로 CDN에 대한 연구가 활발히 진행되고 있다. 본 연구의 목적은 CDN을 통한 디지털콘텐트의 서비스에서 안전하고 효율적인 전송 성능을 지닌 시스템을 설계하는 것이다. 본 연구에서는 CDN을 기반으로 대규모의 디지털콘텐트의 전송 및 관리 시스템에서, 디지털콘텐트의 그룹별 관리를 통한 디지털콘텐트의 안전성과 전송성능을 개선한 시스템을 설계하였다.

1. 서 론

월드와이드웹 사용의 대중적인 증가와 함께 웹에서 디지털콘텐트의 효율적인 전송과 관리가 더욱 중요한 문제로 부각되고 있다. 중앙집중적인 구조를 가진 경우 클라이언트의 급증하는 객체 요청으로 인해 서버시스템에 과부하가 발생할 수 있다. 또한 네트워크 트래픽의 폭주 문제로 인해 객체 전송속도가 급감하여 사용자의 요청에 대한 추가적인 지연이 발생하게 된다. 최근에는 이러한 문제점을 해결하기 위한 디지털콘텐트 전송기술로 CDN(Content Delivery Network)에 대한 연구가 활발히 진행되고 있다[1,2]. 지금까지 많은 연구들이 콘텐트 전송의 성능을 향상시키기 위해 대역폭 증가, 전송의 효율화에 중점을 두었지만, 콘텐트 CDN은 네트워킹 기술을 통해 지능적으로 콘텐트 전송의 성능을 향상시킨 것이다. CDN은 디지털콘텐트의 효율적인 전송과 활용을 위한 기술로 구조적으로는 디지털콘텐트를 제공하는 근원서버(Origin Server)와 클라이언트와 근원서버 사이에 에지서버(Edge Server)를 사용한다. 에지서버는 캐싱 기술을 사용하는 캐시서버를 사용하고 있으며, 캐시 능력은 CDN에서 클라이언트의 요청 처리 능력 및 응답시간에 결정적인 영향을 미치는 요인이다.

또한 원래 인터넷은 보안을 고려하지 않은 전송매체이기 때문에 보안 문제가 발생한다. 일반적으로 웹에서 디지털콘텐트 전송의 안전성을 확보하기 위해 평문을 암호문으로 변환하는 암호화 과정을 통해 전송한다. 이 과정에서, 암호화로 인해 추가적인 부담이 발생하게 되고 암호화된 디지털콘텐트의 전송으로 인한 네트워크 트래픽 증가가 발생하게 된다. 또한 암호화된 디지털콘텐트의 실행을 위해서는 별도의 복호화 과정이 필요하게 되어 클라이언트의 요청 처리에 대한 지연시간을 증가시키게 된다. 처리지연은 네트워크의 물리적인 환경과 암호화 및 암호화된 파일의 전송과 이의 복호화 과정에서 발생한다. 이러한 문제점으로 인해 상용시스템의 경우 mpeg 파일과 같은 멀티미디어 자료에 대한 다운로드 방식에서만 DRM(Digital Right Management) 기술을 통한 암/복호화를 서비스를 사용하고 있으며[3,4], e-러닝이나 영화서비스와 같은 경우 암/복호화 기술을 도입하기에는 구조적인 어려움이 있다.

본 연구에서는 CDN을 기반으로 대규모의 디지털콘텐트의 전송 및 관리 시스템에서, 디지털콘텐트의 그룹별 관리를 통한 디지털콘텐트의 안전성과 전송성능을 개선한 시스템을 설계하였다.

2. 시스템 설계

시스템은 그림 1과 같이 디지털콘텐트를 제공하는 근원서버(Origin Server)와 클라이언트 그룹과 캐시서버로 구성된 에지사이트(Edge Site)로 구성되어 있다. 근원서버와 에지사이트간의 디지털콘텐트의 전송방식은 푸싱(pushing)기법과 캐싱(caching)기법이 있으며, 제안 시스템에서는 객체의 요청에 의해 근원서버에서 전송받는 캐싱기법을 사용한다. 그림 3은 에지사이트의 아키텍처를 나타낸 것이다. 에지사이트는 다수의 클라이언트와 에지서버로 구성된다. 에지서버는 에지서버관리자(Edge Server Manager)와 캐시서버(Cache Server) 및 캐시서버관리자(Cache Server Manager)로 구성되어 있다.

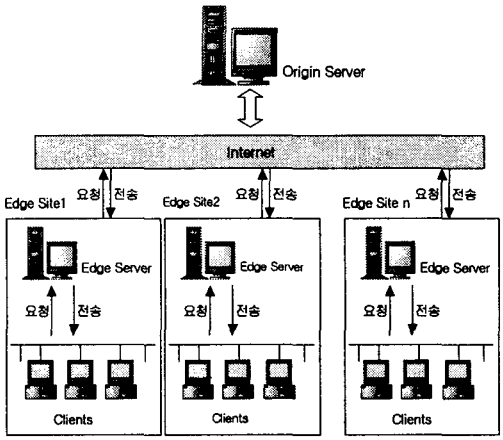

그림 1 시스템 아키텍처

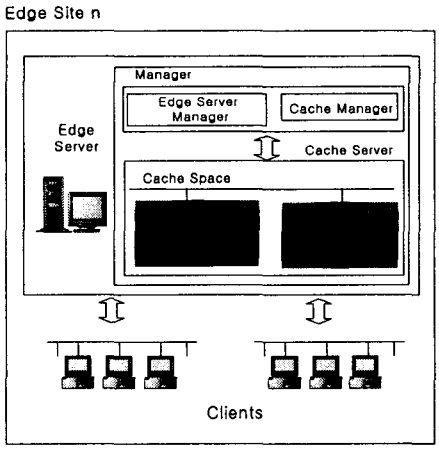

그림 2 Edge Site 아키텍처

2.1 키분배와 DC 전송

에지서버는 클라이언트가 캐시 목록에서 원하는 DC(Digital Content)를 찾을 수 없을 경우 근원서버에서 해당 DC를 전송받아야 한다. 이 경우 근원서버와 에지서버간에는 암호화 데이터를

주고받기 전에 CA(Certificate Authority) 서버를 통해 인증서를 발급 받아야한다. 그림 3은 키 분배 및 DC의 전송 절차를 나타낸 것이다.

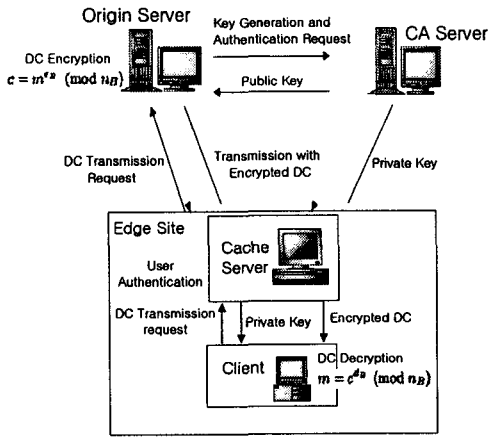

그림 3 키 분배와 DC 전송 절차

c: encrypted message
m: plain message(Content m(0<m<n))
e_B, n_B: public key
d_B, n_B: private key

기존의 RSA 방식에서의 키분배 및 인증 절차를 CDN에서 사용하기 위해서는 다음과 같은 변형이 필요하다. 먼저, CA 서버에 접속한다. 다음으로 CA 서버에 인증서 발급을 요청한다. CA 서버는 인증 요청서를 근원서버와 에지사이트(에지사이트를 구성하는 에지서버)로 전송한다. 근원서버와 에지사이트는 자신의 키 쌍을 생성하고, 인증 요청서를 작성한다. 근원서버와 에지사이트는 자신의 공개키와 인증 요청서를 CA 서버로 전송한다. CA 서버는 수신된 인증 요청서를 확인하여 공개키를 포함한 인증서 발급한다. CA 서버는 근원서버와 에지사이트의 인증 요청서 정보와 인증서를 DB에 저장한다. CA 서버는 인증서를 근원서버와 에지사이트에 전송한다. 근원서버와 에지사이트는 CA 서버로부터 수신된 인증지사이트는 자신의 서를 자신의 비밀키와 함께 저장한다. 에지사이트는 CA 서버로부터 수신된 비밀키를 자신의 에지사이트에서 인증된 클라이언트에게만 전송하고 클라이언트는 이 비밀키를 별도로 저장한다. 또한 각 클라이언트는 에지사이트의 인증을 통해 캐시서버에 저장된 암호화된 DC를 전송받을 수 있게 되어, 별도의 보안절차를 가질 수 있다. 에지사이트에 소속된 클라이언트는 동일한 비밀키를 사용하는 그룹이 되며, 에지사이트의 클라이언트 중에서도 암호화된 DC의 사용 권한에 대한 인증을 받지 못한 경우 비밀키의 전송을 받지 못하며, 암호화된 DC에 대한 권한을 가질 수 없게된다. 결국 CDN의 에지사이트가 DC의 보안을 위한 키관리 및 인증 기능을 가지게 된다. 이것은 대규모의 DC 분배에서 각각의 클라이언트에 대한 관리가 중앙에서 이루어지는 시스템의 경우, 보안성의 확보를 위해 CA 서버 및 근원서버에 발생하는 부담을 줄일 수 있게 한다.

디지털콘텐트의 보호를 위한 암호화 방법은 DES, SEED 알고리즘과 같은 대칭키 방식과 RSA(Rivest-Shamir-Adelman) 알고리즘과 같은 비대칭키 방식이 있다. 이들 기존에 개발된 알고리즘들은 디지털콘텐트의 보호를 위한 각종 요건을 갖추고 있으며, 각종 시스템에 다양하게 응용되고 있다. 대칭키 콘텐트의 암복호화만 가능하여 키값의 교환을 위해서는 Diffie-Hellan(DH)와 같은 별도의 키분배 기법이 필요하게된다. 제안 시스템에서는 키 분배와 암/복호화 및 인증이 가능한 RSA 공개키 기법을 사용한다. 관련 연구에서 다룬 바와 같이 RSA는 현재 가장 많이 사용되는 비대칭 키 기반의 암복호화 알고리즘이며 제안 기법에서 사용하는 RSA의 공개키와 비밀키에 대한 syntax 구조는 표 1과 같다.

표 2 RSA 인증서 Syntax

```
RSAPublicKey ::= SEQUENCE {
        modulus           INTEGER,        -- public modulus, n
        publicExponent    INTEGER         -- public exponent, e -- }

    RSAPrivateKey ::= SEQUENCE {
        version           Version,
        modulus           INTEGER,    -- n
        publicExponent    INTEGER,    -- e
        privateExponent   INTEGER,    -- d
        prime1            INTEGER,    -- p
        prime2            INTEGER,    -- q
        exponent1         INTEGER,    -- d mod (p-1)
        exponent2         INTEGER,    -- d mod (q-1)
        coefficient       INTEGER,    -- (inverse of q) mod p
    }

    Version ::= INTEGER { two-prime(0), multi(1) }
    OtherPrimeInfos ::= SEQUENCE SIZE(1..MAX) OF OtherPrimeInfo
    OtherPrimeInfo ::= SEQUENCE {
        prime         INTEGER,    -- ri
        exponent      INTEGER,    -- di = d mod (ri - 1)
        coefficient   INTEGER     -- ti = (inverse of r1*...*r(i-1)) mod ri
    }
```

인증서는 일련번호, 발행자, 발행일, 만기일, 소유자, 소유자공개키, 발행자서명과 같은 인증에 필요한 기본적인 구조를 포함하고 있다. 인증서 형식 중 발행자와 발행자 서명은 CA와 관련된 필드이다. 이 필드는 인증서의 교환에서 인증서 안의 키가 인증서의 소유자의 것임을 CA가 보장함을 의미한다. 각 인증서는 소유자의 공개키를 포함한다. 그러므로 그 공개키는 인증서의 소유자에게 전달할 데이터의 암호화에 수 있다. 또한 인증서는 발행한 CA의 디지털 서명도 포함한다. 따라서 인증서가 수정되지 않았다는 것과 그 안에 저장된 정보의 신뢰성을 보증하는 것이다.

또한 에지사이트에서는 해당 사이트의 클라이언트와 비밀키 방식의 암/복호화를 통해 안전성을 확보하게 된다. 비밀키 방식은 공개키 방식에 비해 복잡성과 부담을 줄일 수 있지만, 키분배를 위한 별도의 처리를 요구하게 된다. 하지만 제안 시스템에서는 에지서버가 CA서버를 통해 이미 분배받은 비밀키를 클라이언트에서 이용하기 때문에 별도의 키 분배에 대한 부담을 줄일 수 있다.

2.2 에지서버에서 DC 전송과 실행 과정

그림 4는 에지서버에서 클라이언트로 콘텐트의 전송과 실행 과정을 나타낸 것이다.

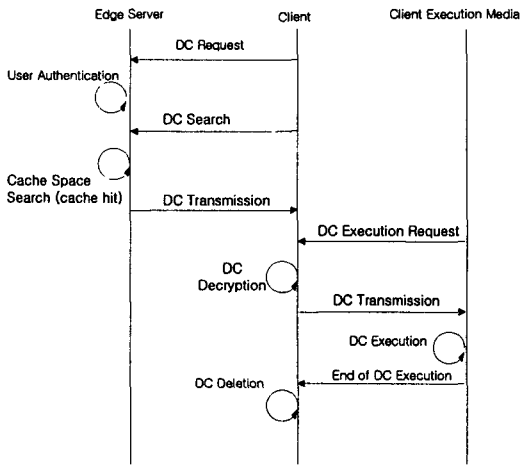

그림 4 에지서버에서 암호화된 콘텐트의 전송과 실행 과정

클라이언트의 DC(Digital Content)를 전송 요청한다. 시스템 자체에서 사용자 인증을 수행한다. 캐시 목록에서 DC 검색한다. 캐시적중(cache hit)일 때 해당 DC를 전송하게된다. 전송 완료 후 클라이언트에서 전송받은 DC의 복호화 수행한다. 키 값에 의한 사용자 인증 절차를 수행한다. 복호화된 DC를 실행한다. 실행을 완료 후에는 클라이언트에서 DC를 삭제한다.

3. 분 석

본 연구를 통의 목적은 CDN을 통한 디지털콘텐트의 전송에서 중요한 이슈는 보안성을 개선하는 것이다. 따라서 제안 시스템의 효용성은 보안성에 관련된 분석을 통해 이루어질 수 있다.

3.1 DC 안전성 분석

비밀키는 공개키에 의해 암호화된 DC를 복호화 하는데만 사용된다. 그러므로 근원서버에서는 중앙의 관리자(CA)로부터 수신자의 공개키를 찾음 다음, 그 공개키를 사용하여 보내는 DC를 암호화한다. 수신자는 그것을 받아서, 자신의 비밀키로 복호화한다. 근원서버에서는 각 에지사이트에 대한 키 값으로 DC를 암호화하게되므로, 각 클라이언트의 공개키로 DC 각각을 암호화하는 부담을 줄일 수 있다.

또한 DC 사용그룹인 에지사이트의 사용자는 각 에지사이트 내에서만 인증되므로, 사용자 인증이 빨라지고 편리해진다. 또한, 제안 시스템에서는 DC의 네트워크 트래픽으로 인한 영향이 감소되고, 에지사이트의 캐시에서 서비스되는 DC의 영향을 받게 된다. 따라서 클라이언트에서의 복호화 시간이 감소되어 실행속도를 향상시킬 수 있다.

3.2 키 분배 및 암/복호화 방식 분석

근원서버에서는 각 에지사이트에 대한 키 값으로 DC를 암호화하게되므로, 각 클라이언트의 공개키로 DC 각각을 암호화하는 부담을 줄일 수 있다. 또한 에지사이트에서는 RSA 공개키 방식으로 암호화된 DC를 복호화한 후, 비밀키 방식으로 암호화하여 별도의 캐시 공간에 관리한다. 비밀키 방식은 공개키 방식보다 지연 요인이 감소할 수 있다는 장점이 있다. 에지사이트 내에서는 공개키 방식보다 복잡성과 부담을 감소시킨 비밀키 방식을 이용하므로 각 클라이언트는 공개키 방식보다 복호화에 따른 지연요인을 줄일 수 있고, 각 클라이언트 각각이 별도의 비밀키를 CA를 통해 분배받아야하는 부담을 없앨 수 있다.

4. 결 론

본 연구에서는 CDN을 기반으로 한 디지털콘텐트의 전송 및 관리 기법에서 안전하고 효율적으로 디지털콘텐트의 전송 기법을 설계하였다.

제안시스템의 추가적인 성능 향상을 위해서는 멀티미디어 객체와 같은 대용량의 웹 객체의 암/복호화에 따른 부담을 줄일 수 있는 방법에 대한 연구가 필요하다. 이는 e-커머스만이 아니라 u-커머스의 DRM 관점에서 복잡성을 줄이고 처리속도를 개선하기 위해 필요하다.

이와 같은 연구를 통해 온라인 교육과 웹 영화, 웹 음악 콘텐츠와 같은 대용량 고품질의 디지털콘텐트를 이용한 ISP(Internet Service Provider)에 활용 가능할 것이다.

참고문헌

[1] 반효경, "CDN을 위한 웹 캐싱 방법", 정보과학회, 제20권 제9호, pp.12-19, 2002.

[2] 최승락, 양철웅, 이중식, "CDN의 핵심 구성 기술들과 경향", 정보과학회, 제20권 제9호, pp.5-11, 2002.

[3] Spctral Lines, "Talking About Digital Copyright," IEEE Spectrum, Vol.38 Issue;6, pp.9, June 2001.

[4] Thorwkrth N. J., Horvatic P., Weis R., Jian zhap, "Security methods for MP3 music delivery," Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on, Vol.2, pp.1831-1835. 2000.

[5] G. Barish, K. Obraczka, World Wide Web Caching: Trends and Techniques. IEEE Communications, Internet Technology Series, May 2000.

[6] H. Bahn, S. Noh, S. L. Min, and K. Koh,

"Efficient Replacement of Nonuniform Objects in Web Caches," IEEE Computer, Vol.35, No.6, pp.65-73, June 2002.

고 일 석

연세대컴퓨터산업시스템공학(박사수료)
미)USIU 경영학과(MBA)
경북대 컴퓨터공학(공학석사)
경북대 컴퓨터공학(공학사)
현재 충북과학대학 전자상거래과 조교수

나 윤 지

충북대 컴퓨터공학(박사수료)
미)NYIT Communication ART 전공
충북대 컴퓨터공학(공학석사)
경북대 생명공학(이학사)