DNSSEC의 "Chain of Trust" 메카니즘에 대한 분석

김학주*, 윤민우*, 임형진*, 송관호**, 정태명***
*성균관대학교 컴퓨터공학과
**한국인터넷 정보센터(KRNIC)
***성균관대학교 정보통신공학부
e-mail:hjookim@rtlab.skku.ac.kr

Analysis of "Chain of Trust" mechanism for DNSSEC

Hak-joo Kim*, Min-Woo Yoon*, Hyung-Jin Lim*,
Kwan-Ho Song**, Tai-Myung Chung***
*Dept of Computer Engineering, SungKyunKwan University
**Korea Network Information Center(KRNIC)
*** School of Information & Communication Engineering,
SungKyunKwan University

요 약

인터넷의 기반이라 할 수 있는 DNS에 대한 보안 필요성이 대두되면서 DNSSEC(DNS Security)이 연구되고 있다. 이는 공개키 기반의 암호화 알고리즘을 기반으로 신뢰사슬(Chain of Trust) 메카니즘을 이용하여 DNS의 근본적인 취약점인 근원지 인증과 무결성 문제점을 해결한다. DNSSEC의 핵심이라 할 수 있는 신뢰사슬 메카니즘은 시스템의 과부하(Overhead)의 감소를 목표로 하여 여러차례 개선이 이루어지고 있다. 본 논문에서는 DNSSEC의 기본 개념 및 신뢰사슬 메카니즘을 살펴보며 앞으로 어떤 방향으로 개발이 지속될 것인지에 대해 분석한다.

1. 서론

인터넷은 급격히 성장하고 있는 네트워크 환경과 더불어 계속 확장되고 있다. 그러나 최근 세계적으로 DNS 서버에 대한 공격이 이어져 이에 대한 보완과 더불어 DNS 자체에 대한 보안의 중요성도 높아지고 있다.

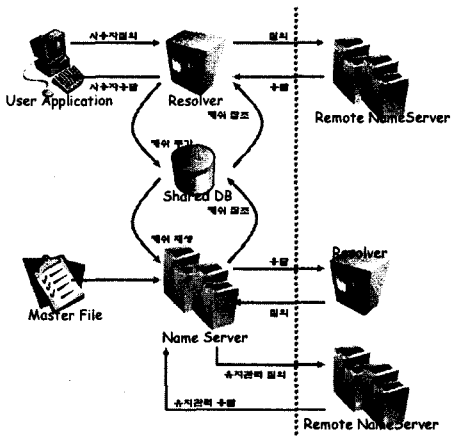
DNS는 인터넷 주소관리의 핵심체계라 할 수 있으며 만일 DNS가 위험에 노출될 경우 대부분의 인터넷 사용자가 인터넷에 접속하기 어렵게 되고 DNS를 이용한 개인정보의 누출 역시 심각한 문제라 할 수 있다. 따라서 DNS 자체에 대한 보안 확장이라 할 수 있는 DNSSEC을 적용하며 암호화 및 복호화에 신뢰사슬(chain of trust) 메카니즘을 사용하여 DNS의 신뢰성을 증대시킨다. 그러나 DNSSEC의 적용은 통신 프로토콜 변경으로 인한 네트워크 부하 증대와 암호화 및 복호화로 인한 시스템의 부하 증대를 야기시켜 DNSSEC 적용에 있어 심각한 장애가 되고 있다. 따라서 본 논문에서는 신뢰사슬 메카니즘을 이해하고 진행되어온 연구내용을 토대로 어떤 문제점이 존재하며 앞으로의 개선방향은 무엇인지에 대하여 논하겠다.

2장에서는 DNS의 일반적인 개념과 DNSSEC이 무엇인지에 대해서 살펴보고 3장에서는 DNSSEC의 핵심이라 할 수 있는 신뢰사슬 메카니즘을 기반으로 하는 암호화 및 인증 메카니즘에 대해 고찰하며 4장에서는 현재 표준화된 RFC2535방식의 신뢰사슬과 현재 연구중인 DS RR을 이용한 신뢰사슬에 대한 고찰 및 비교분석이 이루어진다. 5장에서는 앞으로의 DNSSEC이 나아갈 방향과 목표에 대해 정리하고 추후 연구 계획에 대해 기술한다.

2. DNS와 DNSSEC

2.1 DNS (Domain Name System)

DNS는 IP 주소와 이에 상응하는 계층적인 이름체계를 사상하는 거대한 분산 네이밍 서비스 시스템으로써 트리(tree)형의 분산 데이터베이스 형태를 갖는다. 위임(delegation)과 권한(authority)라는 개념을 이용하여 자신이 권한을 위임받은 존(zone)에 대한 DNS를 구동하며 네임서버와 리졸버(resolver)사이의 질의(query)와 이에 대한 응답(response)을 통해 동작한다[1] DNS의 동작에는 크게 네임에 대한 일반적인 질의와 이와는 반대로 네임에 대한 IP 주소를 얻는 역질의, 존 정보의 동기화를 위한 존 트랜스퍼(zone transfer)등이 있으며 동작 메카니즘으로는 순환질의(recursive)와 직접질의(iterative)방식이 존재한다. 아래의 [그림 1]은 일반적인 DNS의 구조와 간단한 질의 과정을 보여준다[2]



[그림 1] DNS의 구조와 동작

DNS는 인터넷 자원을 관리하며 그에 대한 높은 신뢰성이 요구된다. 그러나 1990년에 벨로빈의 취약성 분석 이후 DNS에 대한 취약점과 이에 대한 보안위협의 연구가 이루어져 DNS 자체에 대한 보안 메카니즘이 요구되었다[3]

2.2 DNSSEC (DNS Secrurity Extensions)

DNSSEC은 DNS의 취약성의 근본 원인이 되는 근원지 인증과 데이터 무결성에 대한 보완책을 적용한 메카니즘으로 DNS 프로토콜의 확장이라고 할 수 있다. DNSSEC에서는 취약점을 해결하고 신뢰성을 유지할 수 있도록 공개키 암호화 메카니즘에 기반한 전자서명(digital signature) 알고리즘을 사용하며 이를 위해 새로이 KEY RR, SIG RR, NXT RR의 세 가지 자원레코드를 정의한다[7]

3. DNSSEC의 신뢰사슬 메카니즘에 대한 고찰

DNSSEC에서의 인증 메카니즘은 초기에 공개키 암호화 메카니즘을 적용한 방식에서 몇가지 문제점이 발견되어 계층구조에서 연계되는 신뢰사슬(chain of trust) 구조로 발전하게 된다. 최근에는 신뢰사슬 구조에 DS(Delegation Signer) RR을 적용한 방식이 제시되고 있다.

3.1 공개키 암호화 기반의 인증

일반적으로 공개키 암호화 방식은 개인키 방식과는 다르게 한 쌍의 키를 이용하여 비동기적으로 암호화 및 복호화를 수행하는 방식을 말한다. 공개키를 이용한 암호화는 DNS에 대한 보안을 위해 초기에 제안되었던 방식으로써 요청을 보내는 리졸버와 그에 대한 응답을 생성하는 네임서버 간에 암호화 및 복호화를 통한 인증이 이루어진다. 이때 네임서버와 리졸버는 1:1 대응관계가 성립되며 네임서버 측에서 생성하는 메시지는 해당 네임서버의 개인키로 암호화되어 요청을 보냈던 리졸버로 전송된다.

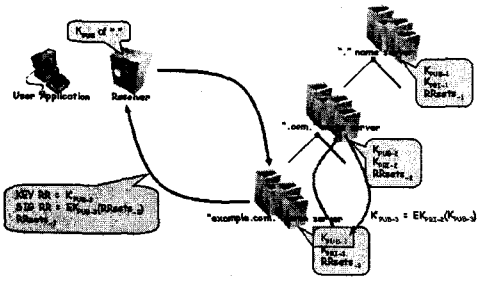
따라서 암호화된 메시지를 복호화하기 위해서 사전에 공개키 분배 과정을 통해 리졸버가 공개키를 소유하고 있어야 할 필요 없이 KEY RR을 이용해 전송된 공개키와 SIG RR을 통해 전송된 암호화된 메시지의 복호화가 이루어진다. 이것은 키 관리 측면에서는 부하가 적지만 전송도중 암호화에 사용되는 키가 유출될 경우 심각한 피해를 입을 수 있으므로 이를 보완할 수 있는 신뢰사슬 메카니즘이 도입된다[5]

3.2 표준 신뢰사슬 방식

RFC2535의 내용에 기반한 표준 신뢰사슬 방식은 기존의 공개키 기반의 인증을 이용하되 해당 네임서버의 공개키를 그 상위의 네임서버의 개인키로 암호화한다는 개념이 추가된 것인데 그것을 그림으로 살펴보면 아래의 [그림 2]와 같다.

여기에서 전제되어야 할 것은 해당 네임서버가 속하는 존의 최상위 네임서버의 공개키가 그 존에 질의를 하게 되는 모든 리졸버에 미리 분배되어 있어야 한다는 것이다. 더불어 실제 질의가 이루어지기 전에 해당 존 내에서는 각 네임서버가 공개키와 개

인키 쌍을 생성하여 공개키를 보다 상위의 네임서버의 개인키로 암호화하는 작업이 수행되어야만 한다. 이렇게 함으로써 전송도중 공개키 정보가 저장된 KEY RR의 내용이 노출되더라도 그보다 상위 존의 공개키 정보가 없다면 KEY RR을 복호화할 수 없기 때문에 공개키의 노출에 대한 취약성이 해결될 수 있다. 이런 작업이 수행된 존을 안전한 존(secured zone)이라고 한다.



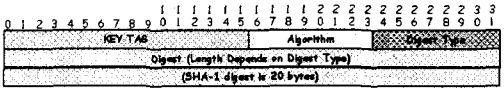
[그림 2] DNSSEC의 신뢰사슬 메카니즘

사전에 안전한 존의 최상위 네임서버의 공개키를 설정하고 있는 리졸버가 해당 존에 어떤 질의를 보내게 될 경우 그에 대한 응답을 보내는 네임서버는 응답 메시지에 포함되는 RRset들에 대해 개인키를 이용한 암호화를 수행하며 이 메시지를 복호화할 수 있는 공개키를 KEY RR을 이용해 보내게 된다. 이 메시지를 전달받은 리졸버는 공개키를 암호화한 개인키를 소유하고 있는 네임서버에 질의를 보내어 KEY RR을 통해 공개키를 전송받으며, 이 과정이 최상위 네임서버까지 이루어지면 자신이 보유하고 있는 공개키를 이용해 순차적으로 공개키들을 복호화해 최종적으로 메시지를 복호화할 수 있게 된다.

이 과정을 통해 DNSSEC에서는 응답을 보낸 네임서버가 안전한 존에 속해있다는 것을 인정할 수 있기 때문에 근원지 인증(source authentication)을 수행하게 되며 복호화된 메시지와 원본 메시지를 비교하여 무결성 검증(integrity validation)을 수행하게 된다[7]

3.3 DS RR을 이용한 신뢰사슬 방식

DS RR은 DNSSEC의 표준화 초기부터 실제 적용에 있어 문제가 있었던 네트워크 부하를 해결하기 위한 하나의 방안으로써 제시된 RR이다. 이 RR을 이용한 신뢰사슬은 기본적으로는 일반적인 신뢰사슬 메카니즘과 동일하지만 하위 네임서버의 공개키를 상위 네임 서버의 개인키로 암호화하는 과정에서 차이가 있다. 아래의 [그림 3]에는 DS RR의 포맷이 나타나 있으며 이 DS RR은 키 태그(key tag) 필드를 이용해 하위 존의 공개키를 가리키게 된다. 따라서 새롭게 위임이 일어날 경우 해당 네임서버에 대한 KEY RR을 상위 네임서버에서 생성하고 이 KEY RR의 태그 값을 저장한 후 그 KEY RR을 하위 존에 전송하기만 하면 된다. 더불어 DS RR을 적용할 때에는 키 관리의 효율성을 위해 하나의 네임서버에 속하는 존에 대한 암호화를 수행하는 존 서명키(ZSK : *Zone Signing Key*)와 키에 대한 암호화를 수행하는 키 서명키(KSK : Key Signing Key) 두 가지의 키를 사용하게 된다[6]



[그림 3] DS RR의 패킷 포맷

4. 표준 신뢰사슬과 DS적용 신뢰사슬의 비교

이 장에서는 앞에서 살펴본 두 가지 신뢰사슬 방식에 대한 동작체계와 부하 비교를 통한 성능 분석을 하고자 한다.

4.1 동작 체계의 비교

표준 신뢰사슬 방식에서는 앞에서도 거론했듯이 하위 존의 공개키를 존 트랜스퍼를 통해 상위 존에 전송한 후 이에 대한 암호화를 수행하고 다시 하위 존에 전송하게 된다. 따라서 하나의 공개키에 대한 설정과정이 최소한 두 번의 전송이 이루어지게 된다. 그러나 DS RR에 기반한 신뢰사슬 방식에서는 상위 존에서 DS RR을 이용해 하위 존에 위임해줄 공개키를 포인팅하고 이것을 하위 존에 전송하게 된다. 따라서 네트워크의 부하는 반으로 줄어들게 되며 ZSK와 KSK를 용도에 따라 나누어 사용하게 되므로 복호화에 두배 정도의 시스템적인 부하가 고려되지만 루트로부터의 공개키의 갱신주기를 따로 설정할 수 있어 공개키 갱신에 따른 네트워크적인 부하는 표준 신뢰사슬 방식보다 상대적으로 줄어들게 된다.

4.2 부하 비교

새로운 위임이 일어나거나 초기의 설정에 있어 키의 생성과 이에 대한 일반적인 복호화 과정에서의 부하는 아래와 같다. 아래의 식에서 N_{KEY}는 KEY 전송으로 인한 네트워크 부하를 나타내며 N_{DATA}는 RRsets의 전송으로 인한 네트워크 부하를 나타낸다. 또한 O_{KEY}는 KEY에 대한 처리를 위한 시스템 부하를 나타낸다.

■ 표준 신뢰사슬 방식

키 설정시의 부하 = $2N_{key} + O_{key}$

복호화 시의 부하

$$= (n-1)N_{key} + N_{key} + N_{data} + O_{key}$$

$$= nN_{key} + N_{data} + O_{key}$$

따라서 전체 부하는,

$$2N_{key} + O_{key} + nN_{key} + N_{data} + O_{key}$$

$$= (n+2)N_{key} + N_{data} + 2O_{key}$$

가 된다.

■ DS적용 신뢰사슬 방식

키 설정시의 부하 = $N_{key} + 2O_{key}$

복호화 시의 부하

$$= (n-1)N_{key} + 2N_{key} + N_{data} + 2O_{key}$$

$$= (n+2)N_{key} + N_{data} + 2O_{key}$$

따라서 전체 부하는,

$$(n+2)N_{key} + N_{data} + 2O_{key} + N_{key} + 2O_{key}$$

$$= (n+3)N_{key} + N_{data} + 4O_{key}$$

이다.

두 경우를 비교해보면 DS를 이용한 신뢰사슬의 경우의 부하가 더 높다는 것을 알 수 있다. 그러나 이것은 일반적인 경우에 대한 부하만을 계산한 것이며 그 외에 빈도수가 잦은 특수한 경우에 대해서도 고려할 필요가 있다. 예를 들어, 표준 신뢰사슬 방식에서는 널 키(NULL KEY)를 사용할 경우, 널 키에 대해서도 일반적인 KEY RR과 같은 처리 과정이 발생하게 되어 이에 대한 처리비용이 발생하는데, 그에 반해 DS RR의 경우 널 키가 사용되는 안전하지 못한 존에 대한 경우 DS RR이 존재하지 않는다는 것으로 그 처리를 대신할 수 있어 처리비용은 표준 신뢰사슬 방식에서의 KEY RR을 처리하기 위한 전체비용을 크게 감소시킬 수 있다.

더구나 DS RR을 사용할 경우에 키를 갱신하는 주기는 ZSK와 KSK가 따로 주어지며 실제 데이터의 암호화에 사용되며 전송되는 KEY RR에 포함되는 내용도 ZSK이므로 KSK의 주기를 RFC2535에서의 키(ZSK) 갱신 주기보다 더 길게 주어도 안전하므로 키의 갱신 및 전파의 횟수를 줄일 수 있어 이로 인해 발생하는 네트워크 및 시스템적인 부하를 크게 줄일 수 있게 된다. 따라서 두 가지 신뢰사슬 방식을 비교해본 결과, 앞으로의 DNSSEC의 방향은 DS RR을 적용한 신뢰사슬의 적용 쪽으로 나아가게 될 것이다.

5. 결론

현재 DNSSEC의 적용을 위한 노력이 계속 이루어지고 있지만 아직도 과부하를 감당할 수 없어 적용에 어려움을 겪고 있다. 따라서 계속적인 과부하의 감소를 위한 방향으로 연구가 진행될 것이며 본 논문에서는 이를 증빙하기 위해 DNSSEC의 핵심인 신뢰사슬의 두 가지 방식을 비교 분석해보았다. 차후에는 기존의 DNS를 바탕으로한 성능평가 인자들과 DNSSEC 적용 시에 새로이 발생하는 성능평가 인자들을 도출하여 이를 감당해낼 수 있는 시스템자원의 요구사항 및 네트워크 요구사항을 구체적으로 습득할 것이며 더불어 프로토콜의 개선을 통해 과부하를 감소시킬 수 있는 방안도 연구할 것이다.

참고문헌

[1] P. Mockapetris, "DOMAIN NAMES - CONCEPTS AND FACILITIES", RFC 1034, November 1987.

[2] P. Mockapetris, "DOMAIN MAMES - IMPLEMENTATION AND SPECIFICATION", RFC 1035, November 1987.

[3] D. Atkins, "Threat Analysis of The Domain Name System", Internet-Draft, November 2002.

[4] Paul Vixie, "DNS and BIND security Issues", Fifth USENIX UNIX Security Symposium in Salt Lake City. Utah, June 1995.

[5] R. Gieben, "Chain of Trust The parent-child and key holder - key signer relations and their communication in DNSSEC", NLnet Labs, November 2000.

[6] Olafur Gudmundsson, "Delegation Signer Resource Record", Internet-Draft, December 2002.

[7] 김학주, "도메인네임 시스템의 취약점 분석과 보안 확장", 한국통신학회지Vol.20 No.7, July, 2003.