DNS의 주소 위변조공격을 막기 위한 IPv6 DNSSEC의 효율성 증대 방안

김형헌*, 박석천**

*경원대학교 소프트웨어학부

Effective Method of IPv6 DNSSEC for DNS Cache Poisoning in DNS

Kim, Hyung Heon*, Park, Seok Cheon*

*Division of Software, Kyungwon University

E-mail : hunniee@lycos.co.kr, scpark@kyungwon.ac.kr

요 약

전 세계적으로 급격한 컴퓨터 보급률 확대와 인터넷 활용인구의 증가에 힘입어 사회와 문화, 산업전반에 걸쳐 인터넷의 활용도는 눈부신 성장을 이어가고 있다. 하지만 기존의 IP 체계가 주소할당과 보안 등의 이유들로 많은 문제점들이 나타나고 있으며 이에 대한 해결 방안으로 지금 전 세계는 IPv6로의 전환에 많은 연구와 노력을 기울이고 있다. 이러한 IP 체계의 전환시점에서도 여전히 발생할 수 있는 DNS 주소 위변조 공격을 막고 안정적으로 도메인 네임 서비스를 사용자에게 제공할 수 있는 방안으로 DNSSEC을 개발하였다. 본 논문에서는 DNSSEC이 가지고 있는 과부하의 문제점을 분석하여 그에 대한 해결방안을 제안한다.

1. 서론

정보통신 기술의 발전과 보급, 인터넷 인구의 폭발적인 증가는 21세기 인류의 문화와 산업 전반에 걸쳐 인터넷을 가장 가깝고 유용한 기술로 만들었다. 하지만 현재 사용 중인 IP 주소 체계인 IPv4는 주소의 개수에서 근래에 부족현상을 갖게 될 것이라는 예측이 나오고 있으며 보안과 비체계적인 주소할당이라는 문제를 안고 있다.

IPv4가 가지고 있는 이러한 문제점을 보완하기 위하여 IPv6가 개발 되었으며 현재는 IPv4/IPv6로의 전환과 관련한 연구와 실질적인 사업 활동들이 활발히 진행 중이다. IPv4의 대안으로 등장한 IPv6는 IPsec(Internet Protocol Security)을 기본적으로 제공하며 플로우 레이블을 이용한 패킷별 품질 제어, 주소 자동 설정 등의 기능이 추가 되었으며 IPv4에 비해 보안기능 및 QoS, 편리성 등의 측면이 대폭 강화되었다. 하지만 이러한 노력에도 불구하고 IPv6가 제공하는 자동설정, 확장헤더 등의 새로운 기능들은 공격자에 의해 악용될 수 있는 보안 위협을 갖고 있다[1].

뿐만 아니라 IPv6가 가지고 있는 이러한 위험들과 함께 현재 사용하고 있는 IPv4 주소 체계에서의 문제점인 DNS(Domain Name Service)에 대한 공격역시 IPv6가 가지는 보안상의 취약부분이다. 특히 전체 인터넷 트래픽에서 매우 많은 부분을 차지하고 있는 DNS 질의, 응답 트래픽의 경우 DoS(Denial of Service)/DDoS(Distributed Denial of Service) 공격과 주소의 위/변조 등의 위험에 노출되어 있다.

본 논문에서는 IPv6 환경에서 발생할 수 있는 DNS 보안 문제를 해결하기 위한 DNSSEC(DNS

* 일반대학원 전자계산학과 석사과정

** IT대학 정교수(교신저자)

Security Extension)을 소개 하고 DNSSEC이 가지고 있는 과부하의 문제점을 분석하고 이에 대한 해결방안을 제안한다.

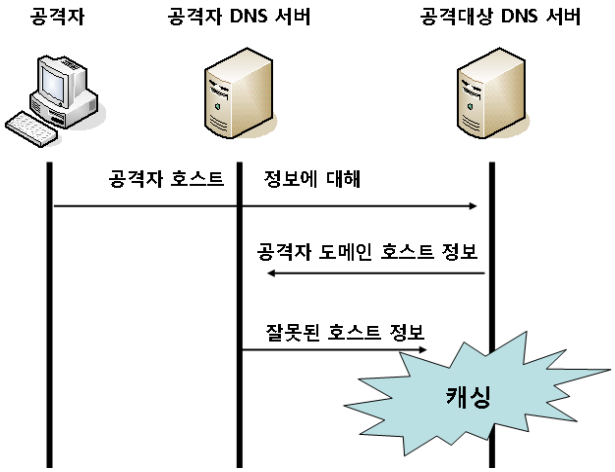
2. 관련연구

2.1 DNS 캐시 포이즈닝(Cache Poisoning)

DNS란 현재 우리가 사용하고 있는 IP 주소체계에서 IP 주소와 이에 대응되는 계층적 이름 체계를 연결 시켜주는 네임 서비스의 구조로써 인터넷 주소 자원 관리의 중요한 역할을 담당하고 있으며 그 자체로써 거대한 분산 데이터베이스이다[2].

이러한 DNS에서 데이터를 위/변조하는 공격 형태를 "DNS 캐시 포이즈닝(DNS Cache Poisoning)"이라 하며, 또는 "파밍(Pharming)"이라고도 한다. 이러한 공격은 공격자에 의해 DNS의 주소가 위/변조되어 오염된 정보를 서버가 클라이언트에게 제공함으로써 사용자들을 공격자가 의도하는 주소로 유인 하거나 서비스거부 공격을 가능하게 한다. 이러한 공격은 검색 결과를 캐쉬하고 있는 Recursive 모드의 DNS 서버에서 주로 발생할 가능성이 있다[3].

그림 2-1은 이러한 공격을 나타내고 있다.



<그림 2-1> Recursive DNS 서버의 캐시 문제

따라서 공격대상 DNS 서버는 검색 결과를 캐쉬하게 되며 이후 해당 도메인의 호스트들은 위조된 캐쉬 결과를 사용하여 인터넷을 사용하게 된다. 이러한 공격을 통해 공격자는 유명 전자상거래 사이트와 똑같은 내용으로 자신의 사이트를 만들어 사용자의 개인 정보를 유출하여 범죄에 악용할 수 있다.

2.2 DNSSEC(DNS Security Extensions)

DNS 표준은 DNS 포이즈닝과 같은 침해공격의 가능성을 고려하지 않던 시대에 정해진 표준 프로토콜이다. DNSSEC은 DNS가 갖고 있는 이러한 보안 취약점을 극복하기 위해 제시된 DNS 확장의 개념으로 디지털 서명과정을 통하여 DNS에서 발생하였던 보안과 관련한 문제들을 극복하기 위하여 정해진 표준 프로토콜이다[4].

DNSSEC은 DNS의 보안상의 취약점을 해결하기 위하여 기존의 프로토콜에 대한 확장이 이루어 졌다. 이는 기존의 프로토콜에 대한 큰 변화를 가져오는 것은 아니며 기존 DNS 프로토콜 자체에 대한 변화는 최소화 시키고 보안 기능을 추가, 향상시키는 방향으로 이루어 졌다.

DNSSEC은 보안 확장을 위해 전자서명과 서명검증 절차를 지원하기 위한 신규 리소스 레코드 RRSIG RR, DNSKEY RR, NSEC RR, DS RR를 정의하고 있으며 DNSSEC 표준은 보안 측면에서 데이터 위/변조 방지가 원천적으로 불가능한 체계로 평가되고 있다[5].

DNSSEC 적용을 통하여 단순히 데이터 위/변조를 방지뿐만 아니라, 일단 DNSSEC의 적용에 의해 데이터 위/변조가 불가능하게 된 도메인 네임 시스템 그 자체가 안전하고 보안 신뢰성이 큰 개방형의 분산구조 데이터베이스의 기능을 유지하게 된다. 이는 보안기능을 가진 응용 어플리케이션 프로토콜에서 필요로 하는 시스템 차원의 시스템 공개키 정보를 안심하고 도메인 네임 시스템에서 리소스 레코드(resource record)로 설정할 수 있게 된다는 것을 의미한다.

2.3 ECC(Elliptic Curve Cryptosystem) 알고리즘

DNSSEC에서 사용할 수 있는 공개키 암호화시스템으로는 소인수 분해의 어려움에 기반을 둔 RSA, RW(Rabin Williams)등과 이산대수문제의 어려움에 근거한 ECC, DSA 등이 있다. 기존의 DLP에 기반한 프로토콜들은 타원곡선의 이산대수 문제가 풀기 어렵다는데 기초하여 타원곡선에서

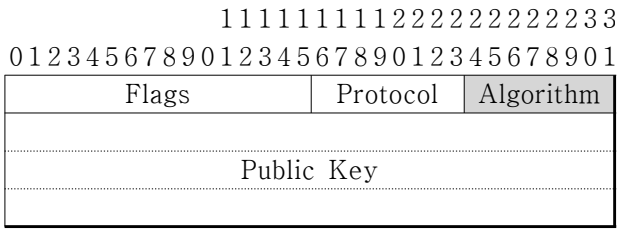
정의된 형태로 변형될 수 있다.

이러한 타원 곡선 알고리즘은 Koblitz[6] 와 Miller[7]에 의해 제안 되었으며 현재까지 subexponential time의 공격방법이 알려져 있지 않아 높은 안전성을 가지고 있다. 특히 ECDSA(Elliptic Curve DSA)의 경우 ANSI X9.62[8]와 IEEE P1393[9] 표준 위원회에서 표준으로 채택되어 이를 방증하고 있다.

뿐만 아니라 ECC는 다른 시스템에 비해 사용되는 키 길이도 현저히 작고(1/6) 연산도 효율적이다. 이와 함께 RSA, DSA 등의 공개키 암호시스템은 ECC 방식보다 큰 길이의 키를 사용하기 때문에 암호 구현 처리 속도가 느리며 RSA는 주요 연산이 곱셈인 반면 ECC는 덧셈이기 때문에 계산이 훨씬 빠르다는 장점이 있다.

3. IPv6 DNSSEC의 효율성 증대 방안

DNSSEC은 DNS 자원레코드 묶음(RRsets)을 서명하고 인증하기 위해 공개키 암호방식을 사용한다. 공개키는 KEY 자원레코드에 저장되고 DNSSEC 인증 과정에서 사용된다. 존은 자신이 권한을 갖고 있는 RRsets를 비밀키를 사용해서 서명하고, 비밀키에 대응하는 공개키를 KEY RR에 저장한다. 모든 경우에 있어서, KEY RR은 안전한 DNS 리졸버 동작(Resolution)과 DNS 메시지 프로세싱에서 특별한 역할을 수행한다. 다음의 그림 3-1은 KEY RR의 포맷을 나타내고 있다.

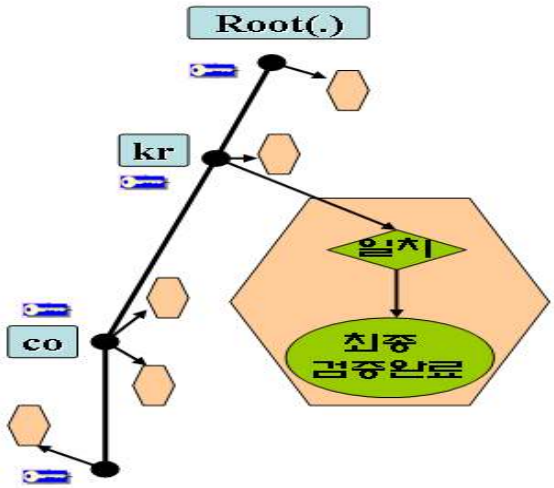


<그림 3-1> KEY RR 포맷

이때 Algorithm 필드의 식별을 통해 암호화알고리즘을 구별하게 된다. 알고리즘의 필드 값을 변경함으로써 다양한 알고리즘을 사용할 수 있는 확장성을 갖는 다는 면에서는 장점이 될 수 있다. 하지만 DNSSEC이 가지고 있는 큰 문제점중 하나인 사용상에서의 과부하를 감안할 때 이를 하나의 안정적인 암호알고리즘으로 고정 시키고 조금이나마 패킷을 줄이는 것이 효율 적이다.

뿐만 아니라 DNS에서와는 달리 KEY RR의 생성과 사용에 있어서 발생할 수 있는 복잡한 경로는 DNS 서비스의 전체적인 질적 저하를 가져 올 수 있다. DNSSEC에서는 키의 생성과 인증의 경로가 증가되고, 이를 처리하는 과정이 복잡하다. 따라서 이렇게 생성되는 키를 관리해줄 특별한 장치가 따로 존재하지 않고 네임 서버, 루트 서버에서 자체 처리한다면 DNS 처리의 과부하와 이로 인한 DNS 처리 속도 저하는 피할 수가 없다.

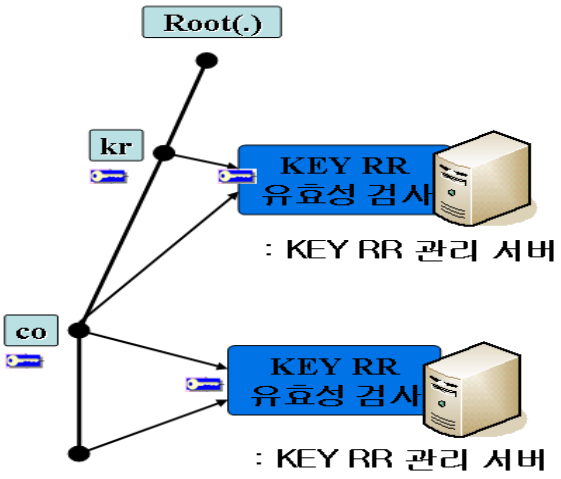
본 논문에서는 이러한 과부하의 문제점을 해결하기 위하여 ECC 암호화 알고리즘을 활용한 KEY RR 처리 서버를 DNS 트리 구조에 추가 할 것을 제안한다.



<그림 3-2> 기존 KEY RR 관리 체계

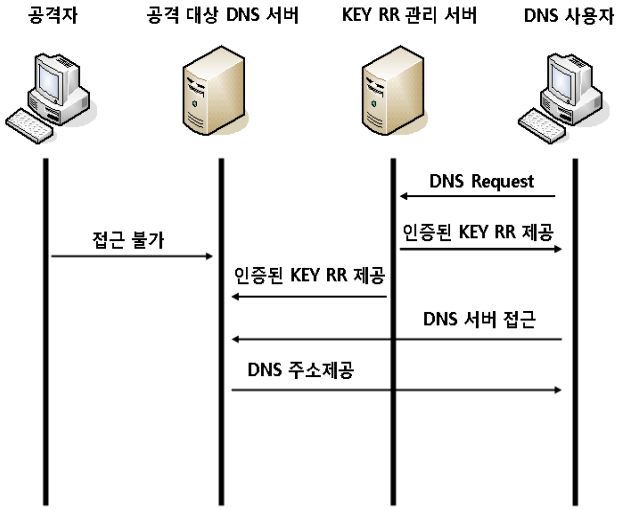
그림 3-2 와 같이 각각의 네임서버에서 처리해야 하는 KEY RR 의 생성을 KEY RR 서버가 직접 관리 하고 이를 처리하는 기능을 대신 한다면 DNSSEC을 처리하는데 소요되는 전체 부하를 줄일 수 있을 것이다.

그림 3-3과 3-4에서와 같이 각각의 네임 서버에서 처리되는 KEY RR 에 관한 작업을 KEY RR 관리 서버에 위임함으로써 KEY 관리에 대한 안전성과 처리의 효율성을 높일 수 있다.



<그림 3-3> KEY RR 관리 서버에 의한 처리

특히 KEY RR 관리 서버는 RSA나 DSA등에서 생성하는 공개키 값보다 더 적은 Public Key를 ECC 알고리즘 이용하여 생성함으로써 KEY의 생성과 검증에 보다 효과적으로 동작할 수 있다.



<그림 3-4> KEY RR 서버에 의한 처리

4. 결론

인터넷 서비스를 이용하는데 가장 핵심 요소인 DNS 서비스는 당연하게도 가장 안정하게 보호받아야 하는 요소이기도 하다. 하지만 DNS 캐시포이즈닝과 같은 공격에 대해 실제로는 완벽하게 안전하지만은 않다. 이에 따라 DNS의 보안을 위해 DNSSEC 와 같은 확장된 프로토콜이 현재 연구되고 적용 활동이 매우 활발하게 진행 중에 있지만 DNSSEC가 확장된 보안효과를 내기 위해서는 앞으로 많은 문제점을 해결해야 할 것이다.

먼저, 인증에 필요한 여러 과정과 새롭게 추가된 KEY RR 과 같은 포맷들을 이용함으로써 발생할 수 있는 처리 데이터양의 증가이다. 뿐만 아니라 보안의 문제를 해결하기 위해 DNS 프로토콜을 확장함으로써 늘어난 레코드를 처리하기 위한 비용 역시 증가할 것이다. 보안이라는 큰 목적을 위해 전반적인 DNS 질의, 응답 시간의 증가는 인터넷 서비스의 저하로 이어 질 수 있는 민감한 문제이다.

이를 해결하기 현재 가장 안전한 빠른 알고리즘으로 인정받고 있는 ECC 알고리즘을 적용하여 KEY RR을 위한 처리 서버를 DNS 트리구조에 추가함으로써 KEY RR을 관리하는데 필요한 다른 서버들의 부하를 줄이고 전체 트래픽을 효율적으로 운용할 수 있을 것이다.

[참고문헌]

[1] E. Davies, “IPv6 Transition/Coexistence Security Considerations,” IETF RFC 4942, September 2007.

[2] Paul Albitz, Cricket Liu "DNS and BIIND 3rd ED," OReilly, 2000.

[3] 유신근, 이현우, “DNS 안전 운용가이드”, February 2001.

[4] D. Eastlake “Domain Name System Security Extensions,” RFC 2535, March 1999.

[5] "DNSSEC의 소개 ", http://dnssec.nida.or.kr/

[6] N.Koblitz, “Elliptic Curve Cryptosystem,” Math, comp, pp203-209, 1987.

[7] V.Miller, "Uses of elliptic Curve in Cryptography," Advances in cryptology - Crypto `85, pp417-426, 1986.

[8] Certicom research, The Elliptic Curve Crypto-system," Certicom, April 1997.

[9] C. G. Pollman, "XML Pool Encryption," XMLSEC02, USA, pp.1-9, 22, November 2002.