DNSSEC の実現課題と そのトランスポートに関する妥当性検証

力武 健次 [†]　中尾 康二 [†◇]　下條 真司 [⋆]　野川 裕記 [*]

ドメイン名システム（DNS）が関与するセキュリティ攻撃が多発している．これらを抑止するために認証プロトコル DNSSEC など DNS の拡張仕様が提案されているが，運用ネットワークへの展開は進んでいない．本稿では過去の DNSSEC の実現性に関する性能指標と課題について概観し，トランスポート問題が DNSSEC の早期普及において解決すべき最も重要な課題であることを示す．またトランスポート問題の中で IP パケット分割が許されているかどうかが DNSSEC のトランスポートの妥当性検証における鍵となる問題であることを述べ，2 つの DNSSEC ホスト間でのトランスポートを検証する実験的手法を提案する．

DNSSEC Feasibility Issues and the Transport Validation Assessment

Kenji RIKITAKE[†], Koji NAKAO[†◇], Shinji SHIMOJO[⋆], and Hiroki NOGAWA[*]

Domain Name System (DNS) has become one of the major targets of the network security incidents. DNS extension protocols such as DNSSEC for the authentication have been proposed, but they are still not widely deployed on the operational networks. In this paper, we survey the past works of the performance metrics and issues on DNSSEC feasibility, and present that the transport issues are the most significant ones to be solved for a faster DNSSEC deployment. We further investigate the transport issues and claim the IP fragmentation allowance is the key issue for validating the transport capability of DNSSEC. We propose experimental methods to validate the transport in between two DNSSEC hosts.

1 Introduction

Domain Name System (DNS) has been one of the critical subsystems of the Internet. Since the traditional DNS has no mechanism to assure the authenticity of the RRs, criminal forgery of DNS RRs and DNS servers becomes popular. Many DNS extension protocols have been proposed to amend the weakness of the traditional DNS. DNSSEC [1, 2, 3] which is intended to provide cryptographic authentication of an RRset [*1], representing the authorizing hierarchy of the global DNS zone delegation. EDNS0 [5] is another extension of the UDP transport usage of DNS, which enables the exchange of larger payloads over UDP datagrams. DNSSEC requires EDNS0 support.

The deployment of DNSSEC, however, is still

[†] 独立行政法人 情報通信研究機構 インシデント対策グループ / Network Security Incident Response Group, NICT, Japan
[◇] KDDI 株式会社 技術開発本部 情報セキュリティ技術部 / Information Security Department, KDDI Corporation
[⋆] 大阪大学 サイバーメディアセンター / Cybermedia Center, Osaka University
[*] 東京医科歯科大学 情報医科学センター / Information Center for Medical Sciences, Tokyo Medical and Dental University

[*1] An RRset is a set of RRs with the same label, class and type, but with different data [4].

slow even though it has long been promoted and discussed in Internet Engineering Task Force (IETF) *dnsext* and *dnsop* Working Groups (WGs). It is still unclear to say that how much overhead DNSSEC will impose on the actual Internet traffic, as of August 2006.

In this paper, we review the feasibility studies for an effective DNSSEC deployment, analyze the transport issues, and propose a validation method of the lower-layer transport for a stable DNSSEC operation. In later sections, we first discuss the general performance issues on DNSSEC deployment to the operational networks in Section 2. We then analyze the transport issues in Section 3, and propose a validation method in Section 4. Our conclusions and future works are presented in Section 5.

2 DNSSEC performance metrics showing the feasibility

Since DNSSEC requires additional computation and data exchange for authenticating the existing RRs, evaluation of the performance issues is needed to estimate the deployment on the operational networks. For example, three key per-host metrics are important for the (authoritative) server, (non-authoritative) cache and resolver operations: memory size, CPU time, and communication bandwidth.

Obtaining metrics regarding the large-scale server-side operations are also significant on the feasibility assessment of DNSSEC operation, such as zone-signing, cryptographic random number generation for the signing keys, and the behavioral change of DNS transport by the introduction of DNSSEC.

The following metrics obtained by the recent measurement and simulation works indicate that they will *not* practically affect the DNSSEC operation by appropriate computing resource allocation.

2.1 CPU time

Processing DNSSEC RRs requires validation of signature, additional payload construction and parsing, and more processing time on dealing with increased number of IP packets. Recent studies show the factor of CPU usage will only be likely to increase by a factor smaller than three. The increase of this factor can be covered by distributing workloads to multiple DNS servers and caches.

Schlyter [6] reported that under some scenarios BIND [7] 9.3.2 only incurred a 2% performance degradation caused by the DNSSEC signature validation measured by the number of processed queries/second, comparing to the traditional (non-DNSSEC) DNS response processing.

Kolkman [8] also reported the CPU usage of DNSSEC authoritative servers only increased from 4~5% of the traditional DNS servers on the simulated environment of a real-world traffic model.

Ager et. al. [9] also reported the CPU time usage on the authoritative servers increased only by a factor of 1.1~2, and on a validation cache the factor was about 2.3.

2.2 Memory size

Processing DNSSEC RRs requires keeping more RR data, of DNS-specific RR types [2] including DNS public key (DNSKEY), RR signature (RRSIG), Next Secure (NSEC) [*2], and Delegation Signer (DS). While those additional RRs will be likely to increase the memory usage by a factor of two to five, this growth will be within the boundaries of current host computer systems.

Ager et. al. [9] reported that the an authoritative server for 55K level-two DNS hierarchy zones used 290Mbytes (MB) of memory footprint for DNSSEC instead of 156MB for the traditional DNS on an Athlon XP1800+ machine running Debian GNU/Linux. The report also showed that a cache processing the workload of 218K queries spread over the complete zone with the TTL values of 24 hours consumed 432MB for DNSSEC instead of 93MB for the traditional DNS.

2.3 Zone signing

On DNSSEC, the server zone data must be previously signed for the validation. Growth of zone data will be well-within the limitation of the available commodity hardware, when we consider the two points tha the full-signing of a zone data is proportional to the fourth order of the key length, and that the signed zone data will become larger than the unsigned one by a factor of four to five. The signing speed will be increased by allowing

[*2] NSEC3 [10] RR specification discussion is ongoing on IETF *dnsext* WG to prevent prohibited zone data copying and enumeration, which is capable by NSEC RRs.

incremental and parallel signing.

The signing experience of `.ca` zone [11] shows the following metrics for the 612k RRsets contained in the zone data as of December 15, 2005:

- Zone data size was increased by a factor of 3.7 when the key size was 1024 bits, from 63MB to 238MB. It became 300MB when the key size was 1584 bits.
- On a Pentium III 1.4GHz machine with 3GB of memory and running Debian GNU/Linux with a 2.6 kernel, it took about 29 minutes to full-sign the zone data when the key size was 1024 bits. It took 85 minutes when the key size was 1584 bits. The estimated full-sign time required is $O(n^4)$ where n is the key size.
- Incremental key signing by splitting up the zone and allowing different expiration date for each subzone took smaller amount of time to complete the signing, provided introducing parallelism. When splitting up the zone into five pieces, each subzone data could be signed in about six minutes. The total processing time would be 9.5 minutes, including the splitting and regeneration processes.

2.4 Random number generation bandwidth

The signing keys of DNSSEC should be well-randomized and cryptographically strong [12]. DNSSEC requires Key Signing Keys (KSK) and Zone Signing Keys (ZSK). The operational requirement of randomness or the generation rate which has to be supplied for signing zones will be far below the maximum capability of the current technology of random number generation.

Kolkman and Gieben [13] suggests the following numbers for the size and effectivity periods of KSKs and ZSKs as follows, from an operational point of view:

- a reasonable key effectivity period for KSK is 13 months, and for ZSK it is a month;
- KSK size should be 1024~2048 bits; and
- ZSK size should not be 100-bit smaller than KSK.

We estimated the frequency of zone signing using the case of `.com` domain, one of the largest global top-level domains. According to a statistic [14], on September 5, 2006 the number of `.com` domains were about 213 million, including the active and deleted second-level domain names. Signing the 213 million zones in 28 days means about 88 ($\simeq 213 \times 10^6/(28 \times 24 \times 60 \times 60)$) zones/sec.

If a zone-signing activity for each zone consumes 1kB (8k bits) of randomness, the randomness required for signing the `.com` domain will be 88kbytes/sec. This rate is well-below the known generation rate capability of random number generator, which is about 10~100Mbps for a cryptographically-safe randomness generator [15].

3 DNSSEC transport issues

Some metric indicates that the *communication bandwidth and the packet/datagram size issues* will largely affect DNSSEC operation performance and should be further investigated. The issues include the consumed bandwidth and UDP payload size for each DNS message, TCP fallback from UDP, and the IP fragmentation. We notified the issues in a previous paper [16], and have found other studies addressing the same issues as well.

3.1 Bandwidth increase

Recent studies of estimation of the DNSSEC bandwidth shows the increase from the traditional DNS case by a factor of two to five, provided all traffics are legitimate and do not include denial-of-service (DoS) attacks.

NIST DNSSEC Project [17] estimated the bandwidth usage would increase by a factor of 3.3 to 3.6 when 50% of the queries are DNSSEC-enabled, and 4.3 to 4.8 when 90% are DNSSEC-enabled.

Kolkman [8] also reported the bandwidth usage of DNSSEC authoritative servers only increased by a factor of 2 to 3. He also noted that the percentage of DO-bit set queries, which indicate those of DNSSEC-enabled resolvers, would be a primary factor of estimating the impact of DNSSEC deployment.

3.2 Payload size increase

Transferring large-payload UDP packets may cause IP fragmentation and increase the number of IP packets. Recent studies of DNSSEC per-packet UDP payload size estimation based on different models of the traditional DNS payloads shows that UDP payloads which cause IP fragmentation will be largely increased, though the amount varies by applying workarounds of reducing DNS answer

payload size.

In our previous paper in 2005 [18], we concluded about 30% of the payloads would not fit in an unfragmented IPv6 UDP packet of 1280-byte MTU, and 15% would not fit in an unfragmented IPv4 packet of 1500-byte MTU, by a payload-length estimation with a real-world DNS traffic data model.

Ager et. al. [9] also reported that the size of 72~77% of the DNSSEC answer payloads without errors or including Name Error responses (NXDomain) would exceed the IPv4 fragmentation limit. Change of protocol from the traditional DNS to DNSSEC requires an NXDomain response to contain the SOA RR, two NSEC RRs with their RRSIGs, and the DNSKEY RRs with RRSIGs.

Kolkman [8] also reported the payload size of DNSSEC answers of BIND 9.3.1 *named* program with modification of assuming all resolvers were DNSSEC-enabled could generate the answer packets which more than 30% of them would exceed the IPv4 fragmentation limit, though in an realistic operation honoring non-DNSSEC resolvers would reduce the percentage to about 10%.

Two workarounds have been proposed to avoid the payload size increase:

- In all of the three research reports mentioned above, it is concluded that *not* including DNSKEY RRs and the related RRSIGs in the additional section will largely reduce the payload length so that the UDP payloads will not exceed the limit imposed by IPv4 MTU.
  This workaround is, however, still not widely deployed. While the *nsd* [19] authoritative server implementation employs this workaround, the *named* of BIND does not.
- In the report of Ager et. al. [9], usage of Elliptic Curve Keys [20] is effective to limit the payload length below the IPv6 MTU limit, though the use of SHA-256 for the DS RRs has been approved instead [21], which is larger than the RSA/SHA-1 [22] RRSIGs currently in use.

3.3 TCP fallback by UDP truncation

UDP payload truncation of 512 bytes due to the traditional DNS specification will cause requerying by TCP, though this is not usually expected on DNSSEC where EDNS0 must be fully supported. A program misconfiguration will cause the same fallback to TCP even in DNSSEC.

Jensen [23] reported, however, that some hosts sent DNSSEC-enabled queries with the upper limit of UDP payload set to 512 bytes, which is the same value for the traditional DNS and is not sufficient for putting in the necessary authority data and the signatures, presumably due to DNS program misconfiguration of *edns-udp-size* of BIND 9. He also reported this behavior immediately showed up as soon as the measured zones were signed. We also noted this TCP-query behavior on a BIND-8-derived resolver [16].

3.4 IP fragmentation of large UDP payloads

DNSSEC specification explicitly requires IP fragmentation [24] support to allow large UDP datagrams to be properly passed through the network. This assumption, however, is not necessarily true due to the end-point firewall and network configuration.

We discussed an actual case of fragmentation-prohibited firewalls in a previous paper [16]. We again experienced a similar case while experimenting DNSSEC-capable resolver program for writing this paper, at a laboratory network of a university. In this case, the packet-filtering firewall of the lab network did not allow fragmented IP packets to get through. We have learned major backbone networks and ISPs allow IP fragmentation, so we suspect the end-point blockage of fragmented packet may become a significant issue on deploying DNSSEC.

4 Transport validation experiments

From the previous discussions of this paper, we found out the most significant issues for guaranteeing the feasibility of DNSSEC is to guarantee both the transfer and forwarding of fragmented IP packets between DNSSEC programs. This has two reasons: 1) IP fragmentation was not necessarily allowed on all operational networks especially on the endpoint firewalls; and 2) for the full transmission of DNSKEY RRs large UDP datagrams are still mandatory, even though the datagram size can be optimized. In other words, we need to *validate the transport* in between the resolvers and servers for passing through the DNSSEC data. We tested

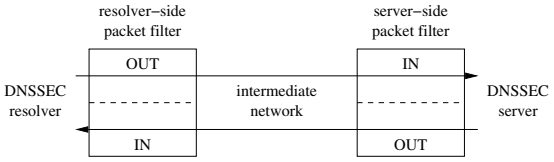

Fig. 1 IP fragmentation and filters between the hosts

two cases for the transport validation.

4.1 Test environment

Figure 1 shows a simple packet filtering model between DNSSEC resolvers (clients) and servers. The filter modules apply two different rules for the outgoing and incoming packets, though usually passing or prohibiting fragmented IP packets are not directionally filtered on the actual filter implementations.

We do not need DNSSEC software for testing the large UDP payloads, since DNS software allowing EDNS0 is the minimal transport requirement for DNSSEC. So we used EDNS0-capable DNS software for testing.

We used a FreeBSD 4.11-RELEASE-p20 host as the resolver, and other public hosts running DNS servers for testing, to safely assume that the server-side filter always passed the fragmented IP packets.

4.2 Requesting large RRs

A large-size RR in a zone can be used for testing the reachability of a large payload on the DNS application software level.

A TXT RR contained 2550-byte RDATA field was made accessible for verifying the large UDP payload reachability in an authoritative zone of a non-recursive DNS server. Sending a large RR had an advantage of *not* being split; a set of multiple RRs would be reduced into a smaller DNS answer containing the partial set.

By sending an EDNS0-enabled query to the TXT RR, the IP fragmentation filtering status could be detected whether the answer RR was received through UDP.

4.3 Sending large UDP queries

Since fragmented-IP filtering is not bidirectional, one can assume if a large UDP datagram were properly received by the server, the filtering would be turned off. The content of a UDP datagram will not be transferred to the upper layer until the fragment reassembly of the IP layer is complete. So if a large fragmented UDP DNS query is properly processed by a DNS server, this indicates the link between the resolver and server allows the IP fragmentation.

Forming a large-size UDP DNS query can be problematic, because the burden of the server should be minimized. So we decided to use a 2048-byte payload, putting the valid DNS query of 45 bytes on the top, of the NS RRs for the Root Zone and the OPT RR enabling EDNS0, and the rest was filled with an arbitrary byte string of 1997 bytes.

We tested a UDP query which included either all fragments or the first fragment only, and found that the fragment reassembly must be completed for a proper query process. Since we could not prohibit the resolver-side outgoing fragments of IP packets, we also had to test the UDP client-server reachability by generating IP packets with the MF (More Fragment) bit [25] with the raw IP socket.

5 Conclusions and future works

In this paper, we surveyed the studies of the performance metrics and issues on DNSSEC feasibility, and concluded that the transport issues was the most significant ones to be solved for a faster DNSSEC deployment. We further investigated the transport issues and found the IP fragmentation allowance was the key issue for validating the transport capability of DNSSEC. We presented two experimental methods to validate the transport in between two DNSSEC hosts.

Our future works include the following three subjects:

- reliability assessment of fragmented IP packets on a large-scale network, and the simulation tool development;
- security implication of allowing large UDP payloads; and
- optimization of DNSSEC server answers to minimize the impact on the Internet-wide deployment.

Acknowledgements

The authors would like to thank Tsunehiko Suzuki for his support on the large-payload DNS

RR field test.

This work was supported by NICT Incentive Research Fund for FY2006.

References

[1] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: DNS Security Introduction and Requirements (2005). RFC4033.

[2] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: Resource Records for the DNS Security Extensions (2005). RFC4034.

[3] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: Protocol Modifications for the DNS Security Extensions (2005). RFC4035.

[4] Elz, R. and Bush, R.: Clarification to the DNS Specification (1997). RFC2181.

[5] Vixie, P.: Extension Mechanisms for DNS (EDNS0) (1999). RFC2671.

[6] Schlyter, J.: DNSSEC Validation Performance Testing. Presentation at the 65th IETF dnsop WG, March 2006, http://www3.ietf.org/proceedings/06mar/slides/dnsop-0.pdf.

[7] Internet Software Consortium: BIND. http://www.isc.org/bind/.

[8] Kolkman, O. M.: Measuring the resource requirements of DNSSEC. RIPE Document RIPE-352, 29 September 2005, http://www.ripe.net/ripe/docs/ripe-352.html.

[9] Ager, B., Dreger, H. and Feldmann, A.: Exploring the Overhead of DNSSEC. April 2005, http://www.net.informatik.tu-muenchen.de/~anja/feldmann/papers/dnssec05.pdf.

[10] Laurie, B., Sisson, G., Arends, R. and Blacka, D.: DNSSEC Hashed Authenticated Denial of Existence. INTERNET-DRAFT draft-ietf-dnsext-nsec3-07.txt.

[11] R. Gieben: .ca Signing Metrics. NLnet Labs document 2006-001, http://www.nlnetlabs.nl/downloads/ca-reg.pdf.

[12] Eastlake 3rd, D., Schiller, J. and Crocker, S.: Randomness Requirements for Security (2005). RFC4086.

[13] Kolkman, O. M. and Gieben, M.: DNSSEC operational guidelines (2006). INTERNET-DRAFT draft-ietf-dnsop-dnssec-operational-practices-08.txt, approved as an RFC by IESG as of March 2006.

[14] Domaintools.com: Domain Counts & Internet Statistics. http://www.domaintools.com/internet-statistics/.

[15] Matsumoto, M., Nishimura, T., Hagita, M. and Saito, M.: Cryptographic Mersenne Twister and Fubuki Stream/Block Cipher (2005). Cryptology ePrint Archive, Report 2005/165, http://eprint.iacr.org/2005/165.

[16] Rikitake, K., Nakao, K., Shimojo, S. and Nogawa, H.: A Study of DNSSEC Operation and Deployment, *IEICE Technical Report ICSS2006-06*, IEICE, pp. 35–40 (2006).

[17] NIST DNSSEC Project: DNSSEC and its Impact on DNS Performance. https://www-x.antd.nist.gov/dnssec/dnssec-perform.html.

[18] Rikitake, K., Nogawa, H., Tanaka, T., Nakao, K. and Shimojo, S.: An Analysis of DNSSEC Transport Overhead Increase, *IPSJ SIG Technical Reports 2005-CSEC-28*, Vol. 2005, No. 33, pp. 345–350 (2005). ISSN 0919-6072.

[19] NLnet Labs: Name Server Daemon (NSD). http://www.nlnetlabs.nl/nsd/.

[20] Schroeppel, R. C. and Eastlake 3rd, D.: Elliptic Curve Keys and Signatures in the Domain Name System (DNS). INTERNET-DRAFT draft-ietf-dnsext-ecc-key-09.txt.

[21] Hardaker, W.: Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs) (2006). RFC4509.

[22] Eastlake, D.: RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS) (2001). RFC3110.

[23] Jansen, J.: Measuring the effects of DNSSEC deployment of query load. NLnet Labs document 2006-002, http://www.nlnetlabs.nl/downloads/dnssec-effects.pdf.

[24] R. Braden (Editor): Requirements for Internet Hosts – Communication Layers (1989). RFC1122.

[25] Wright, G. R. and Stevens, W. R.: *TCP/IP Illustrated, Volume 2*, Addison–Wesley (1995).