8. Appendix:

8.1 Terminology of DNS

DNS Terminology

To understand the DNS and the DNS-specific recommendations in this document, operators and administrators should be familiar with the following terms:

Resolver: A DNS client that sends DNS messages to obtain information about the requested domain name space.

Recursion: The action taken when a DNS server is asked to query on behalf of a DNS resolver.

Authoritative Server: A DNS server that responds to query messages with information stored in Resource Records (RR) for a domain name space stored on the server. A domain needs at least 2 nameserver, one master, one slave. The slave name server will keep up with master nameserver for the change of configuration of domain.

Recursive Resolver: A DNS server that recursively queries for the information requested in the DNS query.

FQDN: A Fully Qualified Domain Name is the absolute name of a device in the distributed DNS database.

Resource Record: A Resource Record (RR) is a format used in DNS messages that is composed of the following fields: NAME, TYPE, CLASS, TTL, RDLENGTH, and RDATA.

Domain: A domain name and all domain names below it, sub-domains; that is, all domain names ending with the domain name. Domains are delegated from one nameserver to another.

Zone: A database that contains information about the domain name space stored on an authoritative server. A zone is the same as a domain minus all delegated domains.

Resource Record Set: All resource records with the same NAME, TYPE, and CLASS; however, the RDATA is different. A response from DNS is always a complete Resource Record Set (or RRSet in short). An example of an RRSet would be multiple NS records for a respective zone or domain.

Delegation Signer (DS): The DS RR is a DNSSEC record type that is used to secure a delegation for a zone.

Zone Signing Key (ZSK): This is used to sign all the RRSets in a zone.

Key Signing Key (KSK): This is used to only sign the DNSKEY RRSet.

DNSKEY Resource Record: This DNSSEC RR is used to store the public keys that are used to sign the records for a zone. This RR can contain either a ZSK or a KSK.

EDNS: Extension Mechanisms for DNS (EDNS, as defined in RFC 2671) is an IETF specification written to remove DNS message size restrictions initially imposed (RFC 1035 Section 2.3.4. Size limits) on the DNS protocol. EDNS allows clients to advertise their capabilities to DNS servers and one of those capabilities that is related to DNSSEC is the ability for a client to advertise its reassembly buffer to a DNS server, for example, a DNS client can reassemble a DNS message sent over UDP that is larger (2000 bytes) than the legacy limit of 512 bytes.

DNSSEC OK (DO) EDNS header bit: This is a bit in the EDNS header that, when set to one ("1") in a DNSSEC-enabled query messages, indicates to the server that the resolver is requesting and able to accept DNSSEC RRs in the query response message.

8.2 Trust of chain

For DNSSEC to work, the recipient needs to know that the public key in use is trustworthy. The resolver asks the name server for its public key, but that public key is used to verify its own identity, which isn’t very secure or verifiable.

To resolve this problem, a "chain of trust" is established. The chain starts by adding an "anchor" at the root name servers. Then each "link" in the "chain" is

signed against the previous "link." Here is our example using www.comp116.edu, which is an A record.

www.comp116.edu is signed at the nameservers for comp116.edu;

comp116.edu is signed by the TLD servers for .edu;

.edu is is signed by the root nameservers.

An anchor for .edu is stored on the root nameservers in the form of a DS (Delegation Signer) record.

8.3 helpful scripts for network manager

Each time you edit the zone by adding or removing records, it has to be signed to make it work. So you can create a script for this so that we don't have to type long commands every time.

```
root@master# more /usr/sbin/zonesigner.sh

#!/bin/sh
PDIR=`pwd`
ZONEDIR="/etc/bind" #location of your zone files
ZONE=$1
ZONEFILE=$2
DNSSERVICE="bind9" #On CentOS/Fedora replace this with "named"
cd $ZONEDIR
SERIAL=`/usr/sbin/named-checkzone $ZONE $ZONEFILE | egrep -ho '[0-9]{10}'`
sed -i 's/'$SERIAL'/'$(($SERIAL+1))'/' $ZONEFILE
/usr/sbin/dnssec-signzone -A -3 $(head -c 1000 /dev/random | sha1sum | cut
-b 1-16) -N increment -o $1 -t $2
service $DNSSERVICE reload
cd $PDIR
```

Save the file and make it executable.

```
root@master# chmod +x /usr/sbin/zonesigner.sh
```

Whenever you want to add or remove records, edit the `example.com.zone` and **NOT the `.signed` file**. This file also takes care of incrementing the serial value, so you needn't do it each time you edit the file. After editing it run the script by passing the domain name and zone filename as parameters.

```
root@master# zonesigner.sh example.com example.com.zone
```

You do not have to do anything on the slave nameserver as the incremented serial will ensure the zone if transferred and updated.

8.3 screen saves of this thesis

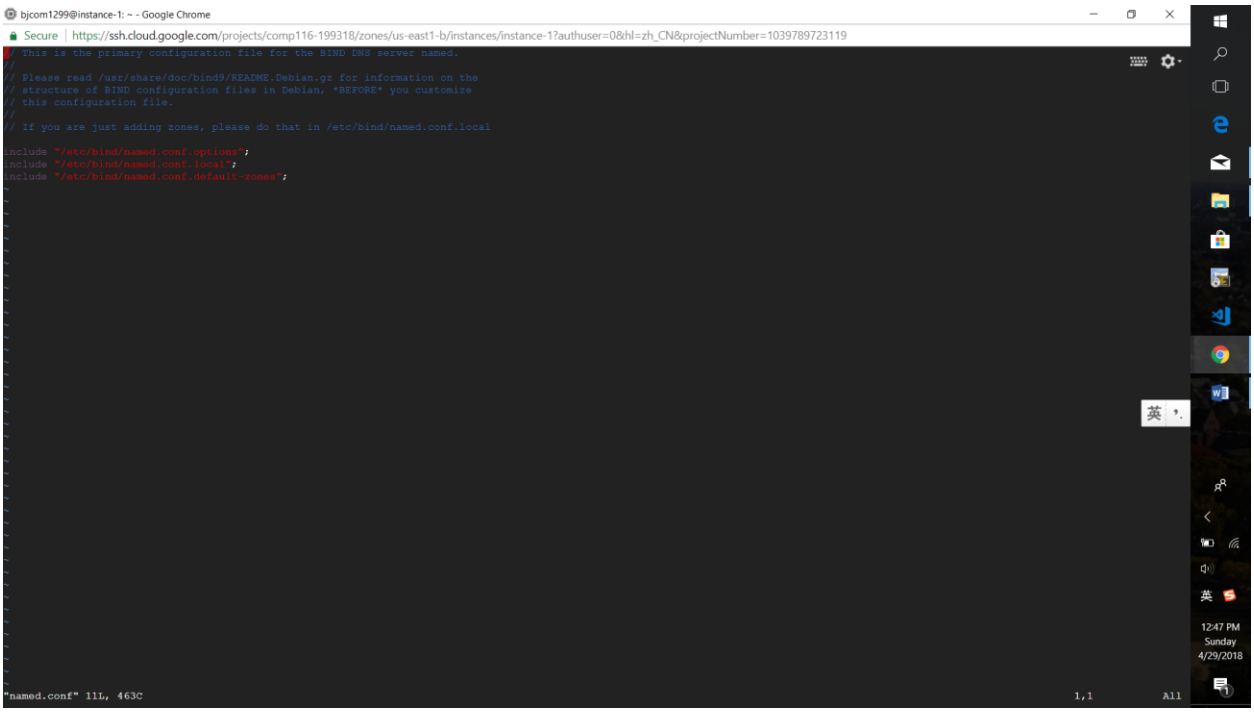

Fig.1 named.conf

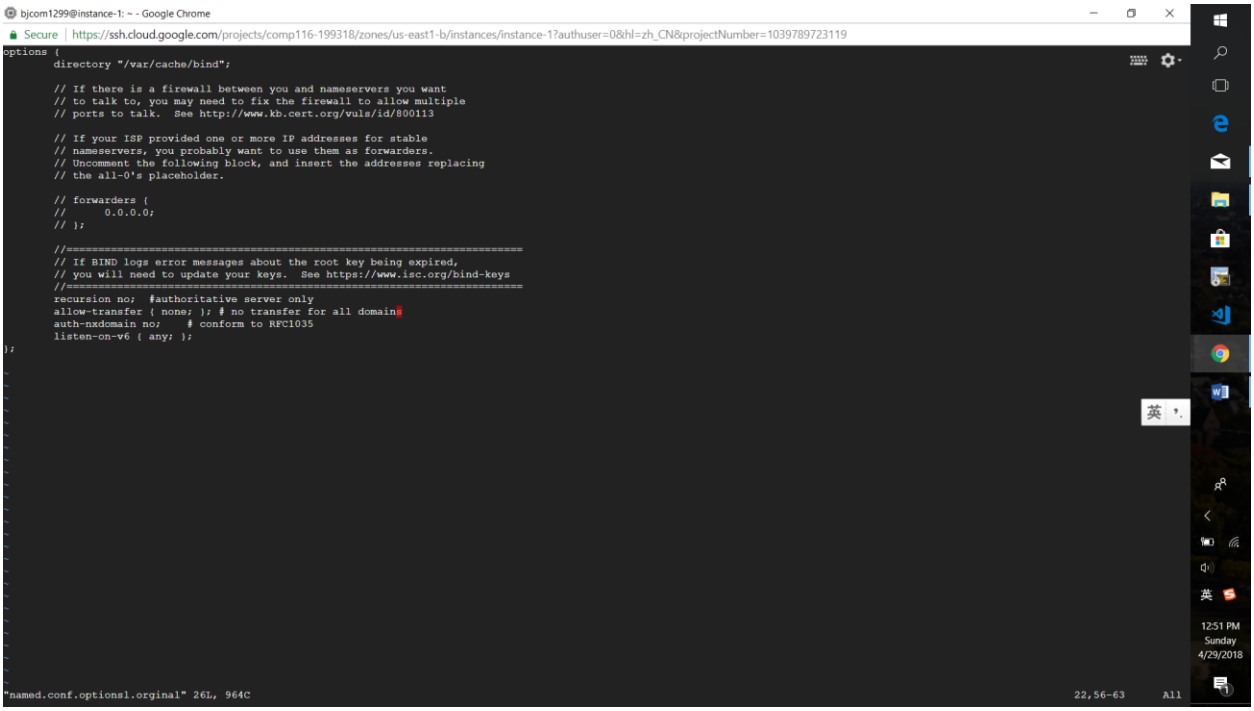

Fig.2 named.conf.options

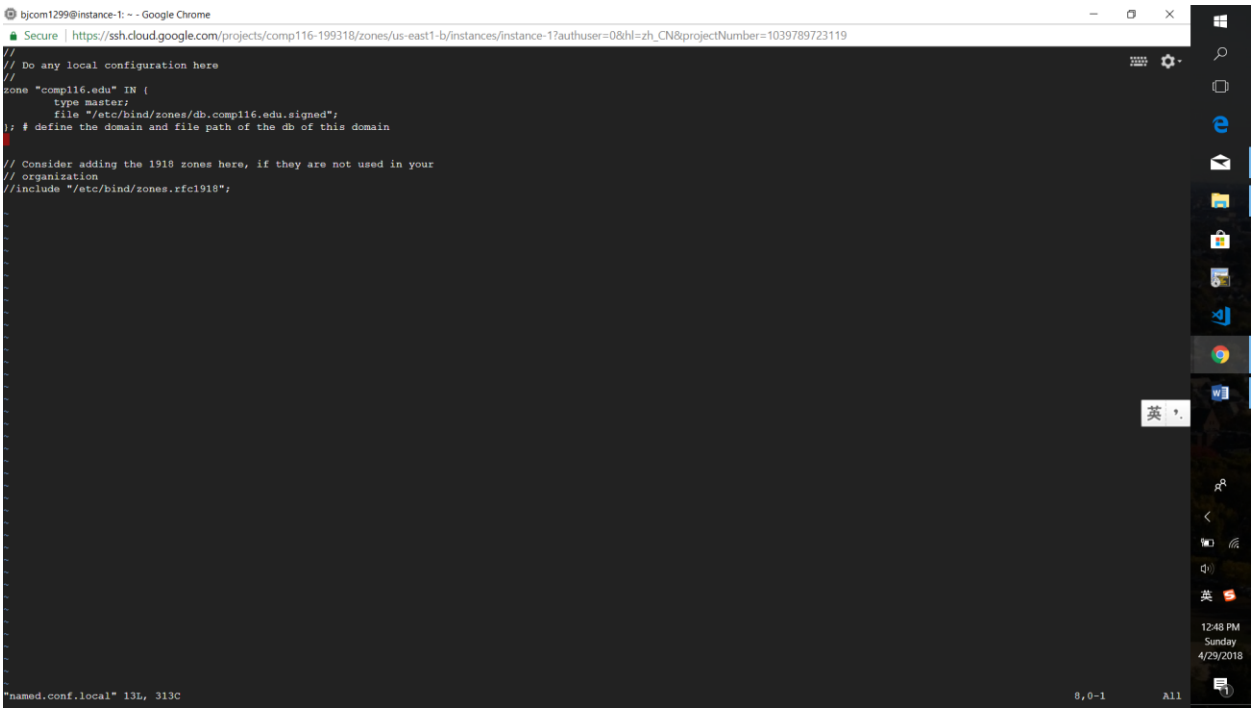

Fig.3 named.conf.local

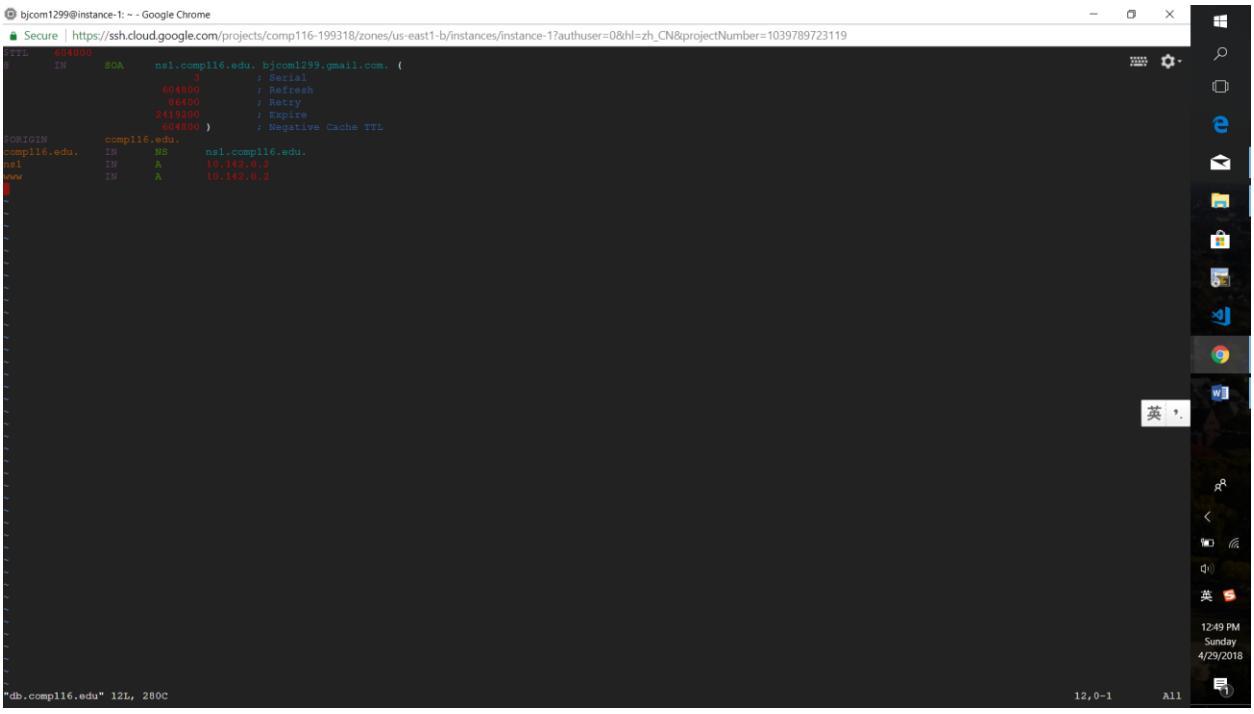

Fig.4 zone file

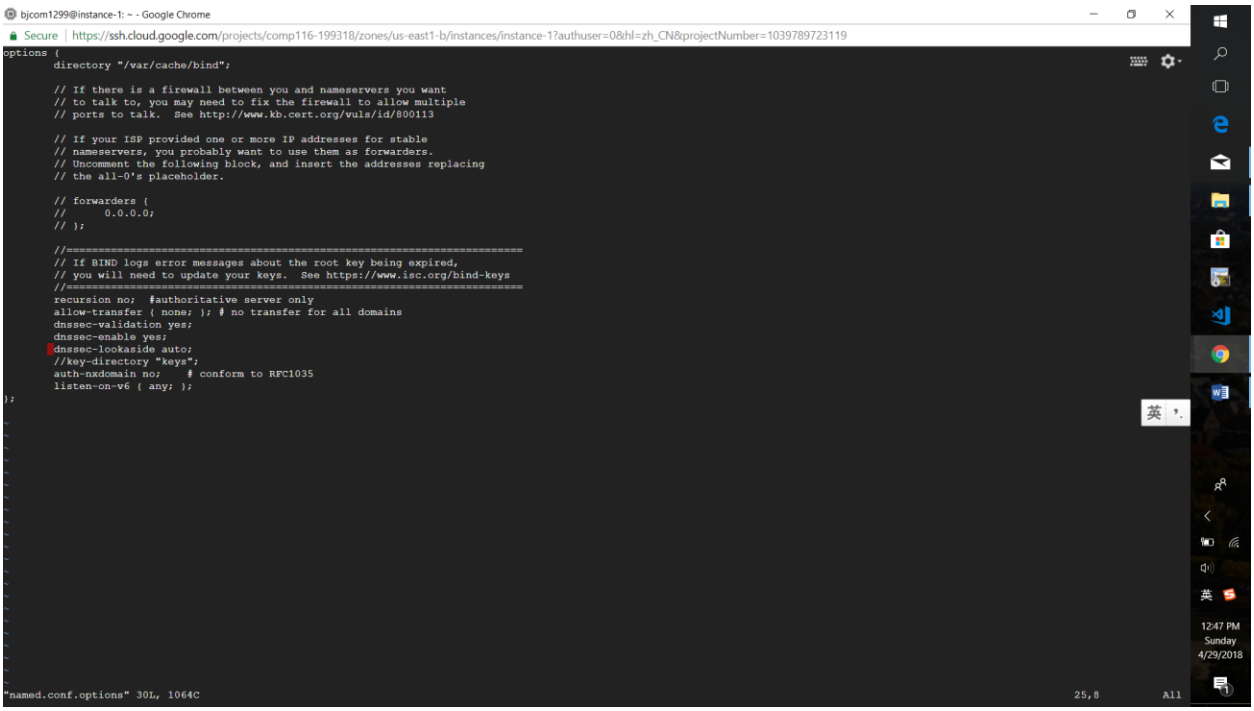

Fig.5 modified named.conf.options (add dnssec)

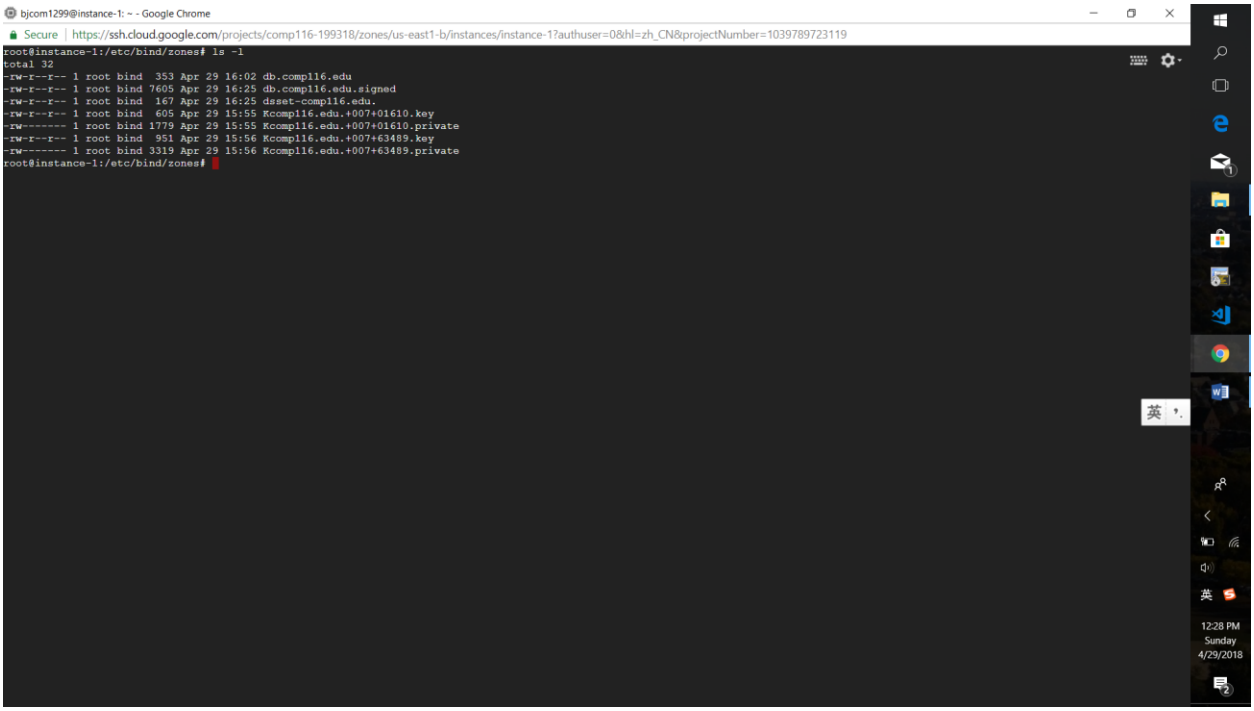

Fig.6 4 keys generated by dnssec-keygen command

How to make your domain more secure: Implementation DNSSEC on your domain immediately

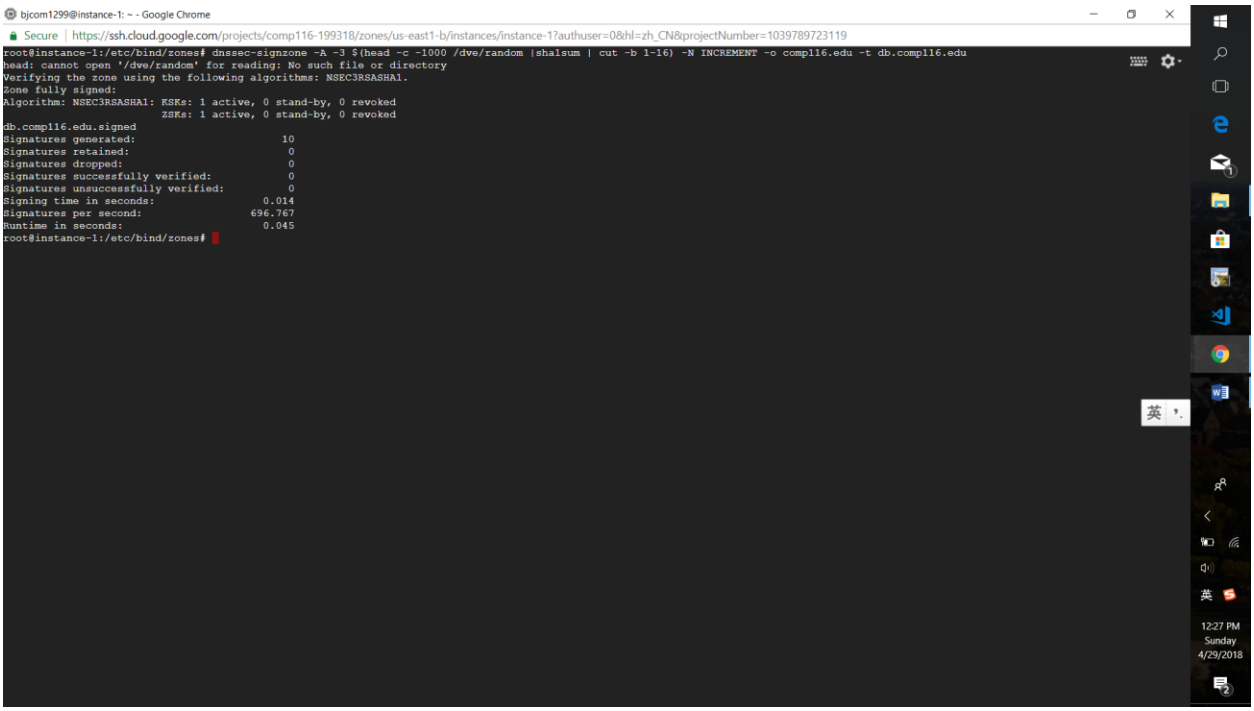

Fig.7 the result of dnssec-singzone command

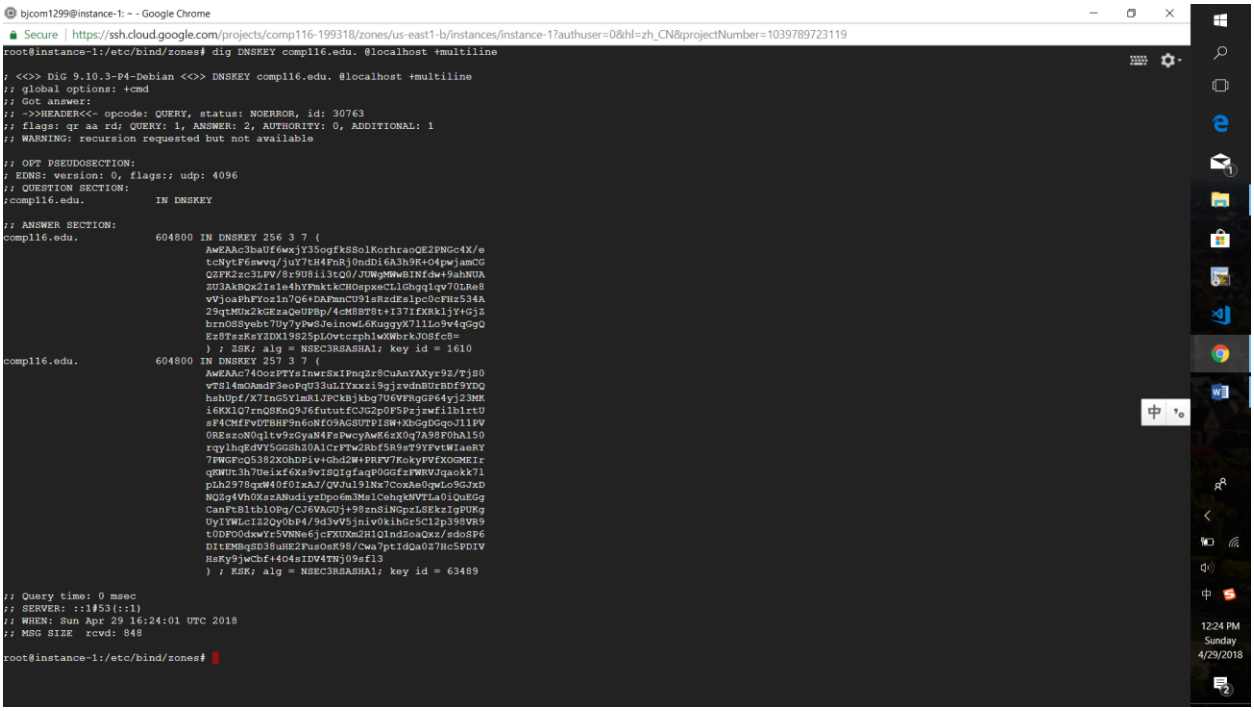

Fig.8 example of dig a signed response

How to make your domain more secure: Implementation DNSSEC on your domain immediately

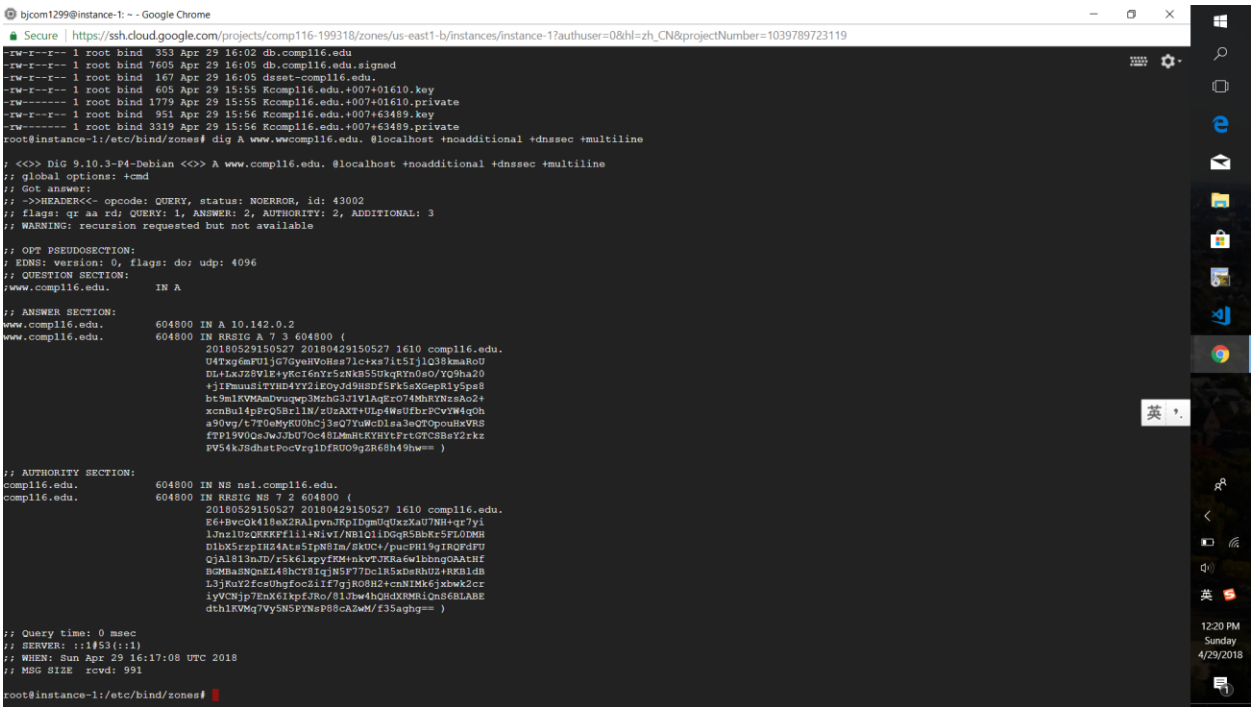

Fig.9 another example of dig a signed response