Ordinary DNS:

Client's Resolver -> www.google.com A?

com. **NS** a.gtld-servers.net
a.gtld-servers.net **A** 192.5.6.30
...

k.root-servers.net
Ordinary DNS:

Client's Resolver → www.google.com A?

com. NS a.gtld-servers.net
a.gtld-servers.net A 192.5.6.30
...

k.root-servers.net

Client's Resolver ← www.google.com A?

google.com. NS ns1.google.com
ns1.google.com A 216.239.32.10
...

a.gtld-servers.net
Ordinary DNS:

Client's Resolver

www.google.com A?

com. NS a.gtld-servers.net
a.gtld-servers.net A 192.5.6.30
...

k.root-servers.net

Client's Resolver

www.google.com A?

google.com. NS ns1.google.com
ns1.google.com A 216.239.32.10
...

a.gtld-servers.net

Client's Resolver

www.google.com A?

...

ns1.google.com
DNSSEC (with simplifications):

Client's Resolver ← www.google.com A? → k.root-servers.net

com. **NS** a.gtld-servers.net
a.gtld-servers.net. **A** 192.5.6.30
...
com. **DS** description-of-com's-key
com. **RRSIG DS** signature-of-that-DS-record-using-root's-key
DNSSEC (with simplifications):

www.google.com A?

Client's Resolver  www.google.com A?

com. **NS** a.gtld-servers.net
a.gtld-servers.net. **A** 192.5.6.30
...
com. **DS** description-of-com's-key
com. **RRSIG** DS signature-of-that-
**DS**-record-using-root's-key

Delegation Signer identifies .com's public key (name and hash)
DNSSEC (with simplifications):

Client's Resolver

www.google.com A?

com. **NS** a.gtld-servers.net
a.gtld-servers.net. **A** 192.5.6.30
...

c. **DS** description-of-com's-key

c. **RRSIG DS** signature-of-that-DS-record-using-root's-key

Retrieving .com's public key is complicated (actually involves multiple keys) ...
DNSSEC (with simplifications):

www.google.com A?

Client's Resolver

k.root-servers.net

com. **NS** a.gtld-servers.net
a.gtld-servers.net. **A** 192.5.6.30
...
com. **DS** description-of-com's-key
com. **RRSIG DS** signature-of-that-
**DS-record-using-root's-key**

Specifies signature over *another* **RR** ... here, the above **DS** record
DNSSEC (with simplifications):

www.google.com A?

Client's Resolver

com. NS a.gtld-servers.net
  a.gtld-servers.net. A 192.5.6.30
...
com. DS description-of-com's-key
com. RRSIG DS signature-of-that-
  DS-record-using-root's-key

Note: no signature over NS or A!
DNSSEC (with simplifications):

Client's Resolver → www.google.com A?

www.google.com A?

Client's Resolver ← a.gtld-servers.net

google.com. **NS** ns1.google.com

ns1.google.com. **A** 216.239.32.10

...
google.com. **DS** description-of-
googles.com's-key

google.com. **RRSIG DS** signature-
of-that-**DS**-record-using-com's-key
DNSSEC (with simplifications):

Client's Resolver — www.google.com A?

...
www.google.com. RRSIG A
signature-of-the-A-records-using-google.com's-key

ns1.google.com
DNSSEC - Mallory attacks!

Client's Resolver

www.google.com A?

www.google.com. A 6.6.6.6

ns1.evil.com
DNSSEC - Mallory attacks!

Resolver observes that the reply didn't include a signature, rejects it as insecure.
DNSSEC - Mallory attacks!

www.google.com A?

www.google.com. A 6.6.6.6
www.google.com RRSIG A
signature-of-the-A-record-using-evil.com's-key

ns1.evil.com

Client's Resolver
DNSSEC - Mallory attacks!

(1) If resolver didn't receive a signature from .com for evil.com's key, then it can't validate this signature & ignores reply since it's not properly signed ...
DNSSEC - Mallory attacks!

(2) If resolver *did* receive a signature from .com for evil.com's key, then it knows the key is for evil.com and not google.com ... and ignores it.
DNSSEC - Mallory attacks!

Client's Resolver

www.google.com A?

www.google.com. A 6.6.6.6
www.google.com RRSIG A
signature-of-the-A-record-using-google.com's-key

ns1.evil.com
DNSSEC - Mallory attacks!

If signature **actually** comes from google.com's key, resolver will believe it …

… but no such signature should exist unless either:

1. google.com *intended* to sign the RR, or
2. google.com's private key was compromised
% dig -_resolver berkeley.edu
Anticipated IPv6 Traffic Growth

- 55% 20% Growth Expected
- 12% 40% Growth Expected
- 12% None, we do not plan to expand IPv6 traffic
- 1% 60% Growth Expected
- 1% 80% Growth Expected
- 9% 100%+ Growth Expected
- 12% Other

Figure 51 Source: Arbor Networks, Inc.
IPv6 Security Concerns

- **72%** Traffic Floods/DDoS
- **57%** Inadequate IPv4/IPv6 Feature Parity
- **53%** Misconfiguration
- **47%** Visibility, I Cannot See the Data Today
- **46%** Stack Implementation Flows
- **45%** Botnets
- **36%** Host Scanning
- **22%** Subscribers Using IPv6 to Bypass Application Rate Limiting
- **4%** Other

*Figure 52 Source: Arbor Networks, Inc.*
IPSec in ESP Transport Mode

Original IPv4 Datagram

<table>
<thead>
<tr>
<th>ver</th>
<th>hlen</th>
<th>TOS</th>
<th>pkt len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>flgs</td>
<td>frag offset</td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>proto=TCP</td>
<td>header cksum</td>
<td></td>
</tr>
<tr>
<td>src IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TCP header (proto = 6)

TCP payload

New IPv4 Datagram

<table>
<thead>
<tr>
<th>ver</th>
<th>hlen</th>
<th>TOS</th>
<th>pkt len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>flgs</td>
<td>frag offset</td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>proto=ESP</td>
<td>header cksum</td>
<td></td>
</tr>
<tr>
<td>src IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ESP

SPI (Security Parameters Index)

Sequence Number

TCP Header + Payload (variable)

Encrypted Data

Authenticated Data

Padding (variable) pad len next=TCP

Authentication Data (optional)

Credit: Steve Friedl