Digital Signatures

- Used to proof authenticity
- Alternative use of a public key cryptosystem
  - e.g., RSA
- Exchange encryption/decryption steps
  - Often used on a message digest
- More details to come ...

Domain Name System (DNS)

- Map
  - Human memorable hostname
    - e.g.: www.net.in.tum.de
  - Machine usable addresses
    - e.g.: 131.159.15.242
- DNS system
  - Application vital to the Internet
  - Transparent to the user
  - Distributed, hierarchical, redundant
  - Uses caching
Need for “secure” DNS

- DNS: no mechanism for authentication
- 1990’s real world attacks:
  - Show how easy misuse of DNS is
  - Attacks:
    - Spoofing
    - Sniffing + answer injection
    - Guessing and predicting query Ids
    - Cache poisoning
  - Bellovin Usenix95: “Using the Domain Name System for System BreakIns”

DNSSEC: “secure” version of DNS

- Developed over the last 10 years
- Goals:
  - full backward compatibility
  - Data integrity
  - Data authenticity
- Problems:
  - Overhead???
    - Bandwidth
    - Resource consumption: Memory, CPU
  - Deployment issues
    - Key management (PKI)
    - Configuration
DNS

- Distributed database
- Hierarchical name space
  - Based on delegation of responsibility

DNS protocol

- Query ↔ Response (1-to-1 relationship)
- UDP (TCP as fallback)
- DNS message
  - Four sections
    - Query
    - Answer
    - Authority
    - Additional
- Resolution process
  - Iterative
  - Recursive
DNS example

- Iterative queries
- Intermediate name servers
  - Delegated zones
  - Query referral
- Information
  - Resource records (RR)
  - Can be cached
- Query types
  - recursive
  - iterative

DNSSEC (Delegation Signer) Basics

- Idea
  - Authentication chain: root zone to resource record (RR)
  - Parent zone guarantees child zone key
- Realization
  - Four new record types
    - RRSIG signatures for resource records (RRs)
    - DNSKEY public key for the zone (e.g., RSA)
    - DS digest of the child-zone’s DNSKEY
      (at delegation points in parent zone
      child’s name equals DS RR’s name)
    - NSEC needed for certifying non-existence
  - Upward compatibility
    - Islands of security
Resolving with DNS/DNSSEC

- Resolving A www.tum.de.: empty resolver cache

  DNS
  Zone: .
  NS: de.

Resolving with DNS

- Resolving A www.tum.de.: empty resolver cache

  DNS
  Zone: .
  NS: de.
Resolving with DNS

- Resolving A www.tum.de.: empty resolver cache

  DNS
  Zone: . de.
  NS: de. tum.de.

A: www.tum.de
Resolving with DNSSEC

Resolving A www.tum.de.: empty resolver cache

DNS/DNSSEC
Zone: .
NS: de.
DS: de.
RRSIG DS: de.

Resolving with DNSSEC

Resolving A www.tum.de.: empty resolver cache

DNS/DNSSEC
Zone: .
de.
NS: de. tum.de.
DS: de. tum.de.
RRSIG DS: de. tum.de.
Resolving with DNSSEC

- Resolving A www.tum.de.: empty resolver cache

DNS/DNSSEC
Zone: .  de.  tum.de.
NS: de.  tum.de.
DS: de.  tum.de.
RRSIG DS: de.  tum.de.
RRSIG A: www.tum.de.
DNSKEY: tum.de.
A: www.tum.de
Resolving with DNSSEC

- Resolving A www.tum.de.: empty resolver cache

DNS/DNSSEC
Zone: . de. tum.de.
NS: de. tum.de.
DS: de. tum.de.
RRSIG DS: de. tum.de.
RRSIG A: www.tum.de.
DNSKEY: . de. tum.de.
A: www.tum.de

DNSSEC problems

- Signing and verification are mathematical complex and need computational power
- DNSSEC packets are larger than DNS packets
  - Larger memory footprint for servers
  - Higher network bandwidth needs
  - Larger packets ⇒
    - Fragmentation
    - Truncation
    - Fallback from UDP to TCP

(DNSSEC requires min pkt size: 1220 Bytes (512 DNS)
recommends pkt size: 4000 Bytes)
DNSSEC overhead

<table>
<thead>
<tr>
<th>Type</th>
<th>Overhead (bytes)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNSKEY</td>
<td>18 + key size</td>
<td>RSA or ECC</td>
</tr>
<tr>
<td>DS</td>
<td>36</td>
<td>SHA-1 digest</td>
</tr>
<tr>
<td>RRSIG</td>
<td>46 + key size +</td>
<td>zone</td>
</tr>
<tr>
<td></td>
<td>70 +</td>
<td>zone</td>
</tr>
<tr>
<td>NSEC</td>
<td>23 +</td>
<td>name</td>
</tr>
</tbody>
</table>

- Typical key sizes in bits:
  - RSA: 1024, 1200 [Kolkman, Gieben, 2004]
  - ECC: 136, 144 [Schroepel, Eastlake, 2004]

Results: DNSSEC overhead

<table>
<thead>
<tr>
<th>Type</th>
<th>Count</th>
<th>DNS size</th>
<th>DNSSEC factor</th>
<th>ECC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>Norm</td>
<td>All</td>
<td>Norm</td>
</tr>
<tr>
<td>Query</td>
<td>32.8M</td>
<td>5.1M</td>
<td>1.5G</td>
<td>0.3G</td>
</tr>
<tr>
<td>noErr</td>
<td>20.0M</td>
<td>4.2M</td>
<td>3.7G</td>
<td>0.7</td>
</tr>
<tr>
<td>Final</td>
<td>6.8M</td>
<td>2.5M</td>
<td>1.2G</td>
<td>0.5G</td>
</tr>
<tr>
<td>Referral</td>
<td>10.9M</td>
<td>1.3M</td>
<td>2.3</td>
<td>0.2G</td>
</tr>
<tr>
<td>Empty</td>
<td>2.2M</td>
<td>390K</td>
<td>.2G</td>
<td>44M</td>
</tr>
<tr>
<td>NXDomain</td>
<td>1.4M</td>
<td>500K</td>
<td>.2G</td>
<td>57M</td>
</tr>
</tbody>
</table>

- Queries: almost no overhead
- Answers: Final and Referral OK but Empty, NXDomain expensive
- Answers: RSA about twice as expensive as ECC
- Answers: All about the same as normalized
### Results: packet size

<table>
<thead>
<tr>
<th>Size ≤</th>
<th>NXDom</th>
<th>noErr</th>
<th>Final</th>
<th>Ref.</th>
<th>Empty</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>1,228</td>
<td>.005</td>
<td>.790</td>
<td>.701</td>
<td>1 .633</td>
</tr>
<tr>
<td>RSA</td>
<td>1,480</td>
<td>.231</td>
<td>.951</td>
<td>.921</td>
<td>1 .996</td>
</tr>
<tr>
<td>RSA</td>
<td>2.056</td>
<td>.999</td>
<td>.979</td>
<td>.991</td>
<td>1 .999</td>
</tr>
<tr>
<td>RSA</td>
<td>4.008</td>
<td>1.000</td>
<td>.999</td>
<td>.999</td>
<td>1 1.000</td>
</tr>
<tr>
<td>ECC</td>
<td>1,228</td>
<td>.998</td>
<td>.999</td>
<td>.998</td>
<td>1 .999</td>
</tr>
<tr>
<td>ECC</td>
<td>1,480</td>
<td>.999</td>
<td>.999</td>
<td>.999</td>
<td>1 .999</td>
</tr>
</tbody>
</table>

- Dig shows default 2056 bytes, 4008 bytes recommended
- 60% of packets not limited by maximum packet sizes (Query, FormErr, ServFail, Refused packets) but less than 1/3 of DNS size and 1/10th of DNSSEC
- noError, NXDomain problematic
  NXDomain: 77% (TR02), 72% (TR04) need fragmentation
- ECC: almost no problem...

### Results: referral pkt size

Shapes of densities similar but shifted (+1 DS RR)
Results: DNS vs. DNSSEC size

- Parallel lines: vertical distance 174 bytes
  - same number of authoritative RRsets

CPU usage increase

- Authoritative name servers
  - Due to larger data volume
  - Average CPU usage (five experiments at 1 M queries)

<table>
<thead>
<tr>
<th>Level</th>
<th>#DNS</th>
<th>CPU-time DNS</th>
<th>#DNSSEC</th>
<th>CPU-time DNSSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5.0K</td>
<td>1.0s</td>
<td>6.4K</td>
<td>2.03s</td>
</tr>
<tr>
<td>1</td>
<td>61.8K</td>
<td>11.56s</td>
<td>65.4K</td>
<td>13.85s</td>
</tr>
<tr>
<td>2</td>
<td>176.1K</td>
<td>23.80s</td>
<td>249.1K</td>
<td>39.71s</td>
</tr>
</tbody>
</table>

- Overhead factor 1.1 to 2
- Overhead factor per query 1.3 to 1.6
**CPU usage increase**

- Caching name servers
  - Due to verification, larger data, stripping of info
  - 218k queries (averaged over 5 experiments)

<table>
<thead>
<tr>
<th>cached</th>
<th>CPU-time DNS</th>
<th>CPU-time DNSSEC</th>
<th>Delay DNS</th>
<th>Delay DNSSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>292.8s</td>
<td>665.4s</td>
<td>1.8ms</td>
<td>3.9ms</td>
</tr>
<tr>
<td>Yes</td>
<td>37.1s</td>
<td>45.9s</td>
<td>0.2ms</td>
<td>0.2ms</td>
</tr>
</tbody>
</table>

- Without caching: factor 2.3
- Absolute additional delay small
- With caching: factor 1.25

---

**Summary**

- Examined cost of wide spread DNSSEC deployment
  - Network bandwidth
  - Resource consumption
- Real world analysis from large client population
- Findings:
  - Increase per packets: avg: 3.4 max: 12.7
  - ECC outperforms RSA
  - Higher memory and CPU requirements

- No apparent show stopper!