Certificate Status using HBS Compression

Phillip Hallam-Baker
Rob Stradling

Problem Statement

- Most common PKI breaches are end entity:
  - Subject breaches ToS
  - Subject discloses key
  - Subject chooses weak key
- Revocation
  - Notify relying parties a certificate isn't trustworthy

Constraints

- 60% of browser use is Open Source
  - Any new browser feature has to be open source compatible.

- Code footprint must be small
- Bandwidth
- Latency
- Third party disclosure

Existing revocation mechanisms

- CRLs
- OCSP (CA)
- OCSP (Stapled)

- Short Lived Certificates
- CRL Sets
  - Just choose the worst of the worst certs

Smaller CRLSets

- SHA-2 hash of revoked cert
  - 256 bits per cert
  - Do we really need every bit?
    - Only 1 million certificates (~2^20)
    - 40 bits should be enough to avoid collisions
  - Don’t need to list revoked certificates
    - Only need to distinguish good certs from bad

Can we do better than 40 bits?

- Yes – can get down to 4 bits per revoked cert.

- Skipping over the details…
  - Time / complexity tradeoffs
  - Encoding overheads

Compressed CRLs

- List of cert hashes:
  - 00 00 00 22 39 ..
  - 00 00 00 4A 20 ..
  - **00 00 00 66 9F .. <REVOKED>**
  - 00 00 00 76 84 ..

EVERY cert with hash **00 00 00 6*** is revoked

Why is TRANS relevant?

- PKIX CRL
  - CA lists bad certs
- TRANS
  - CA registers certs at issue time
- HBS Compressed status sets
  - Do PKIX CRL + TRANS

Practical data sizes

For 2.5 million certs issued, 10% revoked

- Single CRL for all issued certs is 170KB
- Daily Delta CRL is 2-4KB

- These will increase as number of certs issued increases.

Deployment models

- CA Issued
  - Compressed CRLs just a replacement for CRLs
- Single Issuer (e.g. browser provider)
  - Simplifies browser implementation
  - Relies on CAs providing up to date data
  - Probably needs to be based on TRANS

Questions?