Some Lessons Learned from Designing the Resource PKI

Geoff Huston
Chief Scientist, APNIC

May 2007
Address and Routing Security

• The basic security questions that need to be answered are:
  – Is this a valid address prefix?
  – Who injected this address prefix into the network?
  – Did they have the necessary credentials to inject this address prefix?
  – Is the forwarding path to reach this address prefix an acceptable representation of the network’s forwarding state?
  – Can I trust my routing peer / customer / transit ISP to deliver me accurate information?

• Can these questions be answered reliably, quickly and cheaply?
A Resource Validation Framework

• To use a framework to support validation of attestations about addresses and their use

• Queries made within this validation framework should include
  – the authenticity of the address object
  – the authenticity of the origin AS of an advertisement
  – the explicit authority from the address holder to the AS holder that permits an originating routing announcement from that AS
  – the authenticity of the AS path information representing reachability to the address object. i.e. is the next hop address a valid forwarding action for this address prefix?
• As usual, there is no shortage of potential technologies that could conceivably support such a validation framework
  – Certificate Extensions
  – Attribute Certificates
  – Internet Routing Registries++
  – Signed bindings
  – Signed reports
  – The DNS
Design Principles for a Validation Framework

- Don’t force any party to claim to be authoritative beyond its actual authority and knowledge
- Use existing standards
- No new organizations in novel trust roles
- Leverage existing roles and authorities
- Don’t ignore existing processes and functions
- Offer incremental improvements to existing work procedures
- Allow highly reliable and trustable outcomes to be achieved efficiently
What is a Public Key Infrastructure?

- Public/private key pairs can be used for encryption and digital signatures.

- Digital signatures can be used to validate the integrity and authenticity of a message.
  
  By using the public key, I can confirm that the message has not been tampered with and the message was originated by the owner of the matching private key.

- The integrity of the signature validation depends on the knowledge of the public key owner.

- A public key is just a bit sequence.
  
  But:
  
  - **WHOSE** bits?
  - **WHERE** can these bits be used?
  - **WHEN** can these bits be considered valid?

- A **Public Key Infrastructure** is intended to answer these questions.
PKI Rooted Hierarchy

- Explicitly avoid various forms of web of trust models, and use deterministic uniform validation methods based on a combination of issuer subject chains and resource extensions
- Exploit and mirror address allocation hierarchy
  - Each CA in the hierarchy can only validly make attestations and generate certificates about resources that have been delegated to them from the parent CA in the hierarchy
  - Exploit existing authoritative data regarding resource distribution
Modelling the Environment

• Use an **X.509 + PKIX certificate hierarchy** aligned to address distribution points
• The certificate “topic” is the resources allocated from the issuer to the subject at this distribution point
• Certificates allow for the generation of subordinate certificates at delegation distribution points
• Validation of a certificate entails a backwards walk towards the root of the distribution hierarchy
• Revocation can model the return of a resource prior to the termination of the current certificate’s validity period
The Resource PKI
The Resource PKI

It's not just another technology project
- Requires organizational, procedural and legal inputs
- Draws upon many skills to design
- Highly complex space
Issues

• Certificate Requests and Issuance
• Identification of the parties
• Retrieval of certificates by Relying Parties
• Validation of Signatures
• Revocation of Certificates
• Trust Anchor Models
Properties of the Resource PKI

What is the intended use case for this RPKI?

- Validation of attestations about rights-of-use and title?
  - On-demand intermittent single signature validation
  - Can tolerate some amount of visible state transition
  - Outcomes are related to supporting a level of confidence
  - Relying parties do not necessarily require high performance from validation

AND / OR

- Validation of routing protocol updates?
  - In the worst case this could require comprehensive validation across the entire RPKI, within very demanding time constraints, by many replying parties at the same time
  - Real time validation performance
  - Limited / no tolerance for invalid transitional states
Resource Certificates

Resources are not necessarily permanently bound to an identity

- I may have a “right-of-use” for a resource today, but not tomorrow
- While most forms of identity-based PKIs have stable certificate products, there is the potential for greater levels of “churn” in resource certificates
- Relying parties need to constantly refresh their knowledge of the current overall RPKI state
- Efficient repository structures may be critical if there are ~20,000 independent publishers and ~300,000 products to sync against constantly
Certificate Revocation Lists

- Often regarded as the weakest part of the X.509 framework
- CRLs must be issued regularly, must be kept up to date and must be available to relying parties
- Preventing access to a CRL is one of the weaknesses of the RPKI
  - Leads to false positives in validation
- CRLs are used whenever a party no longer has a “right-of-use” over a resource
  - Issue a new certificate with a smaller resource set
  - Revoke the previous certificate

- Design question:
  - Must a CRL be signed with the same private key that was used to sign the certificate that is being revoked?
Certificate Revocation Lists

• Tradeoffs with CRLs and Certificates
  – Smaller validity intervals
    • Reduce CRL size
    • Increase certificate issuance loads
    • Less stable certificates
  – Longer validity intervals
    • CRL bloat
    • More stable certificates
Repository Model

• How do you publish certificates and digitally signed statements?
  – Simple publication process
  Or
  – Ease of use by relying parties for validation

• Single repository model?
  – Critical single resource
  – Potential single point of failure of the entire RPKI
  – Issues of object name uniqueness
  – Issues of management of access control

• Multiple repository model?
  – Each CA publishes in its own repository
  – Issues of name persistence in backward and forward pointers in certificates
  – More complex operations for maintenance of local certificate cache by relying parties
Repositories and Relying Party Access

• How to reference published certificates?
  – In this case it’s a URL
  – A URL with what access method?
    • How many access tools does an relying party need to have at hand?
    • What is the optimal case for access?
      - Fast object retrieval
      - Efficient retrieval of altered objects
    • Optimise for access operations for the server or the client?

• Vulnerabilities
  – Can detect attempts of third party alteration and insertion
  – What about third party disruption by denial?
    • Should the access channel be protected?
      - What are the overheads?
    • Should a repository include a manifest as well as a CRL?
      - A signed list of what should be available in the repository
      - What happens in a denial attack on the manifest?
      - Are there “manifest” PKI standards?
The Identity Bootstrap Question

How does an issuer know that they are certifying the same party as the resource recipient?

– Good question!

– The “its magic” option
  • Somehow, somewhere, sometime in the past, some form of entity-based trust relationship based on key exchange was established between resource issuer and resource recipient
  • This can then be used to establish a key to validate the certificate request as coming from the same entity as the resource recipient
Trust Anchor Models

• What / who are the trust anchors for this RPKI?
  – *Standard answer:* the choice of trust anchors is made by a relying party as a local configuration task
  – *In practice,* proposed Trust Anchors are provided with the distribution of relying party toolkits
    • IE: Tools -> Internet Options -> Content -> Certificates -> Trusted Root CAs
    • Trust anchors should be (relatively) stable
  – *Pragmatic answer #1:* the root of the resource distribution hierarchy: IANA
    • But what if we get into a DNSSEC-styled impass over signing at the “root” of the hierarchy?
  – *Pragmatic answer #2:* Use RIR-issued self-signed certificates as trust anchors with delegated resources
    • But these certificates will change as blocks are passed to the RIRs (i.e. monthly!)
    • So how can this regularly updated trust anchor material be distributed to all potential relying parties?
Key Rollover

Is hard!

• How quickly can you re-issue all subordinate certificates with the new key?
  – How far down the hierarchy do you need to re-issue?

• How quickly can you revoke products signed with the old key?

• Are there intermediate states that create unintentional invalidity of signed products?
Digitally Signed Products

- How can you “revoke” an authority granted through a signed authority document?
  - Signed objects are not certificates
    - No lifetime
    - No CRL
    - No …

- Propose to use “one-off” keys and end-entity certificates
  - Generate a key pair
  - Generate an end-entity certificate for this key pair
  - Publish the certificate
  - Sign the object with the private key
  - Destroy the key pair
What have we learned so far?

- There’s an entirely new terminology universe in the X.509 certificate space!
  - Dark Rites of Initiation into the security world appear to be necessary!

- X.509 certificate specifications appear to include a vast repertoire of extensions with elastic semantics
  - choose carefully!

- There is limited PKI deployment experience out there
  - each PKI development exercise is a learning experience

- Distributed authority models are very challenging to design in a robust manner
  - Think carefully about the model of synchronization across a realm of multiple issuers and multiple repositories with dynamic authoritative information
What have we learned so far?

• Resource Certificates are a means to an end, not an end in and of themselves
  – make the certificate work to suit the business model rather than the reverse

• This is not an exercise that is done lightly
  – considerable investment in expertise, tools, documentation, and navel-gazing over process is useful

▪ Outcomes need to represent superior choices for players
  ▪ Risk mitigation is an ephemeral and diverse motive for widespread adoption
  ▪ Better, faster, and cheaper solutions tend to produce better adoption motivations

▪ Good (and Useful) security in a very diverse environment is a very challenging objective
Thank You