STIR Credentials

IETF 88 (Vancouver)
Wednesday Session
Jon & Hadriel
Requirements Discussion

• Trying to identify some of the primary design decisions
• Credentials will require public and private keys
  – How we provision private keys and publish public keys
• Differences between proposed solutions can be elusive
• Let’s level-set and get on the same page
Enrollment

• How do signers get credentials?
  – Delegation from above?
    • May be from regulators, or from other number holders
    • Might be thousand blocks, or individual numbers
  – Proof of possession?
    • Per Whatsapp, VIPR, etc.
    • Weaker assertion, but still useful enough?

• Credential strength a critical dimension of this
  – Is there any usability story for weak credentials?
  – “Golden root” versus “silver sprouts”
    • National-level delegation roots
Req: Delegation

• Much discussed on the list
  – Premise: anyone with a credential for a number may delegate to someone below them
    • Non-exclusive, may delegate to multiple parties
  – Delegation may be all of the delegators authority, or only part of it
    • If I have a thousand block, I can give you 999 numbers or just 1

• Temporary delegation
  – One-time use
  – Doctor’s office case
  – Call centers
  – Need accountability for temporary delegation
Req: Credentials for Ranges

• Some entities will have authority over multiple numbers
  – Administrative domains could control millions of numbers
    • In non-continuous ranges
  – Includes service providers, enterprises, resellers, etc.
  – Some entities will only have one number

• Ideally, a service provider should not have to have one credential per number
  – Expressing those ranges is an important decision here
Expiry, Revocation and Rollover

• All credentials will have a lifetime
  – Caching expressed as a TTL or similar lifetime indicator
  – Numbers change owners, get ported, transfer normally

• Sometimes keys will be compromised before their expiry
  – Some sort of real-time checking required
    • DNS could set TTLs very low
    • OCSP checks, but with some overhead
    • Are these two forms of overhead equal?
Signer Provisioning

• How do signers acquire and manage private keys?
  – Self-generated and provisioned at the authority
  – Generated by the authority and downloaded to devices

• Intermediaries and enterprises
  – Provision keys for number blocks, sign on behalf of calls/texts passing by
  – May possess many keys

• End user terminals
  – Built into the device?
  – Downloaded from the authority?

• In both cases, may need keys for the same authority range provisioned in multiple places
Verifier Credential Acquisition

• Different methods of acquiring credentials
  – Push (credential arrives with the request)
    • Caching unlikely
  – Pull (verifier acquires credential on receipt of request)
    • Either dereferencing a URI or creating a fetch based on the originating number
    • With caching
  – Prefetch (verifier gets top 500 keys) with pull
    • Maybe pub/sub service
  – Others? Probably
Which credentials do verifiers need?

• Can we uniquely identify the needed credential based on TN alone?
  – Depends on how many authorities there are

• How many authorities and delegates per number?
  – Some kind of hint needed to disambiguate
    • Identity-Info
    • CIDER “public key index value”
Public or Confidential Database?

• How much information are we willing to make public?
  – Will we reveal the carrier of record?
    • Okay when a call is received to know the originating carrier?
      – Receiving user vs. receiving carrier may be different
    • More seriously, can an attacker mine a public database to reveal who owns all numbers?
  – Will we introduce VIPR-like privacy leaks

• If we make the database where verifiers get credentials confidential, how limiting will that prove?
  – How important is endpoint verification?
    • Does trust become transitive if endpoints rely on intermediary verifiers?