DKIM Key Update

John Levine
IETF 98
March 2017
What does DKIM do?

- Message header and body hashes
- Signature added to message
- Recipient system verifies

Private key in signer
Public key in DNS
Current algorithms

• Hash algorithms
  • SHA-1 (deprecated, little used)
  • SHA-256

• Signature algorithm
  • RSA

• Algorithm choice
  • Signature says what hash and sig algorithms
  • Key says what sig and allowed hash algorithms
Signers must use at least 1K keys

Verifiers must support 512 to 2K signing keys

Key stored in DNS TXT record
  - 1K key is almost 256 characters
  - TXT records store text in <256 chunks
  - Provisioning crudware often doesn’t handle multiple chunks
  - Yes, this is a stupid problem, but it’s not going to change
Better signatures through crudware

• Add new algorithm
  • Probably ed25519
  • Keys are small
  • Not in OpenSSL yet, will be soon

• Publish key hashes
  • Put the public key in the signature
  • Put a fixed size key hash in the DNS

• Could do either or both
Transition issues

• Sign with old and new until everyone handles new
  • For some version of everyone
• RFC says only one key record per selector
  • Could relax to one per key type
• Verifiers ignore signatures they don’t understand
  • But how reliably do they do that?
  • Probably a lot of poorly tested code there
• ARC is coming
  • Similar to DKIM, would be nice to do it right up front