The Need for Cryptographically Insecure Hash Functions

Eric Rescorla
RTFM, Inc.
ekr@rtfm.com
Cryptographic hash functions are useful... too useful

• Reminder: \( H(M) \rightarrow \{0, 1\}^*b \)

• Used in all sorts of non-security settings
  – Generation of unique fixed-length identifiers [JLR+08]
  – Content “fingerprints” [BWNH+06, SLHbC08]
  – “Strong” checksum [FGM+99]

• These are non-adversarial settings
  – The cryptographic guarantees are not used here

• Disadvantages
  – Performance
  – Confusion
Why this is confusing

- When cryptographic digests are used, people expect them to be security critical
  - Even worse now that MD5 has been weakened
  - Reviewers ask “what about hash agility?” “Where’s the security analysis?”
  - Need to explicitly disclaim security usages

Because the maximum number of inputs which need to be compared is 70 the chance of a collision is low even with a relatively small hash value, such as 32 bits. CRC-32c as specified in [RFC4960] is a specific acceptable function, as is MD5 [RFC1321]. Note that MD5 is being chosen purely for non-cryptographic properties. An attacker who can control the inputs in order to produce a hash collision can attack the connection in a variety of other ways. [draft-ietf-sip-fork-loop-fix-08.txt]
We need standardized insecure hash function(s)

- Can be used instead of cryptographic hashes
  - Faster
  - Explicitly weak
  - Serves as a signal that it’s not security critical
- Requirements
  - Fast
  - Low collision probability: chance of $H(M) = H(M')$ is $2^{-b}$
  - High probability of detecting small errors
  - Easy to find collisions and preimages
- Lots of existing hashes (CRC, universal hashing, ...)
  - Let’s pick one (or two)
References


