RFC5019-bis

https://datatracker.ietf.org/doc/draft-bonnell-rfc5019bis/02/

C. Bonnell, C. Wilson, T. Ito, S. Turner
NIST deprecated the use of SHA-1 in 2011 and disallowed its use for digital signatures at the end of 2013, based on both the Wang et. al attack and the potential for brute-force attack. In December 2022, NIST published the plan to transition away from the current limited use of the SHA-1.

https://csrc.nist.gov/projects/Hash-Functions
Profile for High-Volume Environments

This Draft

I-D 5019-bis

Allow use of SHA-256 as the hashing algorithms for CertID

RFC 5019

MUST use SHA-1 as the hashing algorithm for the CertID.issuerNameHash and the CertID.issuerKeyHash values.”

(General) Protocol Specification of OCSP

RFC 6960

Defined CertID without restriction on hashing algorithm

RFC 6277

Defined CertID without restriction on hashing algorithm

RFC 2560
What do we want for IETF117

• WG adaption
Supplemental information
How to migrate (Basic idea)

• Mandate SHA-256 CertID for new OCSP responders and clients
  • Responder return SHA-256 response for SHA-256 request, SHA-1 response for SHA-1 request
  • Wait for transition

• To reduce signing throughput for pre-generate response,
  • response may include multiple SingleResponse objects for the same certificate, each with a different CertID hash algorithm (as Rob mentioned)
Other approaches

• Respond only with a single algorithm
  • return SHA-1 responses both for SHA-1 and SHA-256 requests
  • Then, Switch to responders, which return SHA-256 responses both for SHA-1 and SHA-256 requests

• We believe, it will damage backward compatibility or people would stick with using SHA-1
Potential Questions

• Why you do not marge to 6960 and make 6960-bis?
  • We believe it is great to make 6960-bis, however,
  • 6960-bis is about general protocol and would take much longer to process.
  • 6960-bis would have many other discussion apart from SHA-1 issue

• Why you do not change responderID also and marge to this draft?
  • We can, but with following reason, we prefer to separate
    • responderID is defined in 6960, and we need to change 6960
    • Migrations of a profile and a protocol tend to be different,
    • Scope of migration / backward compatibility would be much broader for 6960-bis
    • We believe certID would be much more useful to used for other revocation mechanism and had better migrate earlier.