SIP Digest Access Authentication

Rifaat Shekh-Yusef

IETF 89, SIPCore WG, London

March 6, 2014

Algorithms Agility

- **New Algorithms**
  - SHA-256
  - SHA-512/256

- **IANA Registry**
  - HTTP Digest Hash Algorithms Registry

“HTTP Digest Hash Algorithms” Registry

Hash Algorithm	Digest Size	Preference	Reference
MD5	32	1.0	RFC XXXX
SHA-512-256	64	2.0	RFC XXXX
SHA-256	64	3.0	RFC XXXX

Update Policy: Specification Required

Forking

- **Forking Proxy**
  - Aggregates challenges into a single response.
  - Multiple challenges should be differentiated by the **realm**.
  - Multiple challenges might belong to the same **realm**.
    - Can these challenges use different algorithms?

- **UAC**
  - Provides authorization for each **realm** using the top/ preferred algorithm.

Forking Backward Compatibility

	Option 1	Option 2
Resource Proxy	Algorithms in order of preference	Algorithms in no particular order
Forking Proxy	Must maintain order	Order is not significant
UAC	Select the top algorithm per realm	Select the most preferred algorithm per realm, as defined in the IANA Registry.

QoP Backward Compatibility

- **RFC3261, Section 22.4, bullet 8**

  Use of the "qop" parameter is optional in RFC 2617 for the purposes of backwards compatibility with RFC 2069; since RFC 2543 was based on RFC 2069, the "qop" parameter must unfortunately remain optional for clients and servers to receive. However, servers MUST always send a "qop" parameter in WWW-Authenticate and Proxy-Authenticate header field values. If a client receives a "qop" parameter in a challenge header field, it MUST send the "qop" parameter in any resulting authorization header field.

- **RFC2617 -** H (H(A1) | nonce | nc | cnonce | qop | H(A2))
- **RFC2069 -** H (H(A1) | nonce | H(A2))

Feedback?

- Forking
- QoP Backward Compatibility

- WG adoption