Supporting SHA-3 in IETF Protocols and Implementations

Tim Polk
tim.polk@nist.gov
July 28, 2011

Goal

- Goal is to inform the IETF regarding the level of effort that will be required to support the eventual SHA-3 algorithm in IETF protocols, and their implementations*, should the community so choose*

- This is not a report on the progress of the SHA-3 Competition

Acknowledgements

- Material addressing relevant aspects of the competition were stolen from Bill Burr's presentation at Quo Vadis 2011 in Warsaw:
  - "The SHA-3 Competition to Date: Where are we and what have we learned?"
- Protocol impact assessment is based largely on Larry Bassham's experiences with various SHA-3 candidate algorithm implementations

Cryptographic Hash Function

- Hash functions take a variable-length message ***x*** and reduce it to a shorter fixed-length *message digest* ***hash(x).***
- Core requirement: Use ***hash(x)*** as a stand-in for ***x*** in digital signatures, MACs, file comparisons, etc.
- Many applications: "Swiss army knives" of crypto:
  - Digital signatures (with public key algorithms)
  - Message authentication codes & user authentication (with a secret key)
  - Key update and derivation
  - Random number generation
  - One way function
  - Code recognition (list the hashes of good programs or malware)
  - Commitment schemes and random oracles

Overview of Hash Function Standards

- MD5: 128-bits [RFC 1321, '92]
  - Small break in 1996, badly broken in 2004
- SHA-0: 160-bits [FIPS 180, '93]
  - Quickly withdrawn, publicly broken in 1998
- SHA-1: 160-bits, [FIPS 180-1, '95] [RFC 3174, '01]
  - tweak to SHA-0
  - Wang attack in 2005
    - Why don't we have a demonstrated collision by 2011?
- SHA-2: 224, 256, 384 & 512-bit variants, [FIPS 180-3, '03] [RFC 4634, '06]
  - No significant attacks, but…

Reasonable Doubt?

- MD5, SHA-0, SHA-1 all broken
  - MD5, SHA-0, SHA-1 and SHA-2 are all from the same family
- Is SHA-2 next?
  - Given the lead time required to introduce a new hash function into actual use, can we afford to take that chance?

Strategy for an Alternative to SHA-2

- Hold an International Competition, based on the experiences with AES, for a new "Advanced Hash Algorithm" (widely referred to as SHA-3)
- NIST no has plans to withdraw SHA-2
  - Needed to start transition away from SHA-1 based on tag length alone
  - Threat to SHA-2 was conjectural
- NIST wanted SHA-3 to be
  - SHA-2 plug compatible
  - At least as strong as SHA-2 and likely to survive an attack on SHA-2
  - Improve upon SHA-2 if possible

Minimum Acceptability Requirements

- "The algorithm shall be publicly disclosed and available worldwide without royalties or intellectual property restrictions"
- Implementable in a wide range or hardware and software platforms
- SHA-2 plug compatibility
  - Digest sizes of 224, 256, 384 & 512 bits
- NIST reserved the right to change schedule and extend or add rounds

Evaluation Criteria

- Security is the most important factor
  - Must be secure in common hash function applications
    - HMAC, PRF, etc.
  - Collision resistance approx. *n/2* bits
  - Preimage resistance approx. *n* bits
  - 2^{nd} preimage resistance approx. *n-k* bits
  - Any subset of the digest should have similar properties

Evaluation Criteria (cont.)

- Computational efficiency (speed)
  - Software and hardware
- Memory requirements
- Flexibility
  - Tunable parameter
  - Secure & efficient on many platforms
  - Parallelizability
- Simplicity

The Source of Confusion: Tunable Parameters

- Tunable parameters were identified as one mechanism for achieving flexibility
  - "*The algorithm is parameterizable, e.g. can accommodate additional rounds*"
- This seems to contradict requirement for a "plug replacement", since SHA-2 algorithms are not tunable
  - Protocols would need to be altered to convey additional information
- Tuning will be performed before standardization
  - Round counts, etc. will be fixed in SHA-3
  - No additional parameters to pass

SHA-3 Competition Timeline

- ✓ 01/23/07 Proposed criteria for new hash algorithm
- ✓ 11/02/07 SHA-3 Competition announced
- ✓ 10/31/08 Submissions due – 64 received from 191 submitters
- ✓ 12/09/08 Announced 51 first-round candidates, first round began
- ✓ 02/25/09 First SHA-3 Candidate Conference, Leuven Belgium
- ✓ 07/24/09 Announced 14 second-round candidates
- ✓ 09/15/09 Accepted algorithm tweaks, second round began
- ✓ 09/18/09 Published the first-round report (NISTIR 7620)
- ✓ 08/23/10 Second SHA-3 Candidate Conference, UCSB
- ✓ 12/09/10 Announced 5 finalists
- ✓ 01/31/11 Accepted final tweaks, third round began
- ✓ 02/16/11 Published the second-round report (NISTIR 7764)
- ✓ 03/22/12 Final SHA-3 Candidate Conference, Washington, DC
- ✓ Summer 12 Announce final selection
- ✓ 1Qtr 13 FIPS package to Sec. of Commerce

Impact on IETF Protocols

- Depends upon the degree of cryptographic agility in the protocol, and the mechanism used to achieve that agility
- Protocols that have cryptographic agility with respect to hash algorithms should be easy
  - Generally, just specify new code points
    - Semantics will be straightforward
- Protocols that lack cryptographic agility with respect to hash algorithms will be a chore
    - Can't infer anything from the size of the hash value!
  - First, redesign the protocol for hash agility in general
  - Then, assign code points and specify semantics

Impact on Implementations of Agile Protocols

- Implementations that currently support SHA-2 can be easily upgraded to support SHA-3
  - In general, replicate code for SHA-2 of same size
    - Use the new code point
    - call SHA-3 for the hash value
  - If an implementation doesn't support SHA-2 now, implementing now will speed support for SHA-3!
- Implementations that currently support smaller hash values (e.g., SHA-1) may require additional effort
  - Larger buffers, changes in message encodings, etc.

My Two Cents

- For agile protocols, addition of SHA-3 will be straightforward
- For legacy protocols, the vast bulk of the effort will focus on adding agility to the protocol
  - Can't differentiate 256 bit SHA-2 hash from a 256 bit SHA-3 hash on the wire!
- For implementations of agile protocols that support SHA-2, adding SHA-3 will be a breeze
- For implementations that don't support SHA-2 but intend to support SHA-3, implement SHA-2 now and get the bugs out

Questions?