Appendix V Evaluation Criteria for SHA-3

William Stallings

Supplement to
Cryptography and Network Security, Sixth Edition
Prentice Hall 2013
ISBN: 0133354695
http://williamstallings.com/Cryptography

V.1 THE ORIGINS OF SHA-3

As of this writing, SHA-1 has not yet been "broken." That is, no one has demonstrated a technique for producing collisions in a practical amount of time. However, because SHA-1 is very similar in structure and in the basic mathematical operations used to MD5 and SHA-0, both of which have been broken, SHA-1 is considered insecure and has been phased out for SHA-2.

SHA-2, particularly the 512-bit version, would appear to provide unassailable security. However, SHA-2 shares the same structure and mathematical operations as its predecessors, and this is a cause for concern. Because it will take years to find a suitable replacement for SHA-2, should it become vulnerable, NIST decided to begin the process of developing a new hash standard.

Accordingly, NIST announced in 2007 a competition to produce the next generation NIST hash function, to be called SHA-3. The basic requirements that must be satisfied by any candidate for SHA-3 are the following.

1. It must be possible to replace SHA-2 with SHA-3 in any application by a simple drop-in substitution. Therefore, SHA-3 must support hash value lengths of 224, 256, 384, and 512 bits.
2. SHA-3 must preserve the online nature of SHA-2. That is, the algorithm must process comparatively small blocks (512 or 1024 bits) at a time instead of requiring that the entire message be buffered in memory before processing it.

NIST received sixty-four entries by October 31, 2008; and selected fifty-one candidate algorithms to advance to the first round on December 10, 2008, and fourteen to advance to the second round on July 24, 2009. Based on the public feedback and internal reviews of the second-round candidates, NIST selected five SHA-3 finalists to advance to the third (and final) round of the competition on December 9, 2010. NIST completed its evaluation process and announced a final standard in 2012. NIST selected Keccak for the SHA-3 algorithm. Keccak was designed by a team of cryptographers from Belgium and Italy: Guido Bertoni, Joan Daemen,[1] Michaël Peeters, and Gilles Van Assche. In their announcement, NIST explained the choice as follows:

> NIST chose KECCAK over the four other excellent finalists for its elegant design, large security margin, good general performance, excellent efficiency in hardware implementations, and for its flexibility. KECCAK uses a new "sponge construction" chaining mode, based on a fixed permutation, that can readily be adjusted to trade generic security strength for throughput, and can generate larger or smaller hash outputs as required. The KECCAK designers have also defined a modified chaining mode for KECCAK that provides authenticated encryption.

The role of SHA-3 is somewhat different from that of AES. In the case of AES, NIST approved AES as a replacement for DEA and 3DEA. Although 3DEA is still considered secure, it is not efficient and has a smaller key length than one of the AES options. On the other hand, SHA-2 has held up well and NIST considers it secure for general use. So SHA-3 is a complement or alternative to SHA-2 rather than a replacement. The relatively compact nature of SHA-3 may make it useful for so-called "embedded" or smart devices that connect to electronic networks but are not themselves full-fledged computers. Examples include sensors in a building-wide security system and home appliances that can be controlled remotely.

[1] Joan Daemen is one of the two designers of Rijndael, the winner of the AES competition a decade earlier.

V.2 SHA-3 EVALUATION

It is worth examining the criteria used by NIST to evaluate potential candidates. These criteria span the range of concerns for the practical application of modern cryptographic hash functions. When NIST issued its original request for candidate algorithm nominations in 2007 [NIST07], the request stated that candidate algorithms would be compared based on the factors shown in Table V.1 (ranked in descending order of relative importance). The three categories of criteria were:

- **Security:** The evaluation considered the relative security of the candidates compared to each other and to SHA-2. In addition, specific security requirements related to various applications and resistance to attacks are included in this category.
- **Cost:** NIST intends SHA-3 to be practical in a wide range of applications. Accordingly, SHA-3 must have high computational efficiency, so as to be usable in high-speed applications, such as broadband links, and low memory requirements.
- **Algorithm and implementation characteristics:** This category includes a variety of considerations, including flexibility; suitability for a variety of hardware and software implementations; and simplicity, which will make an analysis of security more straightforward.

Table V.1 NIST Evaluation Criteria for SHA-3 (page 1 of 2)

SECURITY

- **Applications of the hash function:** Algorithms having the same hash length will be compared for the security that may be provided in a wide variety of cryptographic applications, including digital signatures (FIPS 186–2), key derivation (NIST SP 800–56A), hash-based message authentication codes (FIPS 198), and deterministic random bit generators (SP 800–90).
- **Specific requirements when hash functions are used to support HMAC, pseudorandom functions (PRFs), and randomized hashing:** The criteria list specific security requirements for these applications.
- **Addition security requirements:** Specific collision and preimage resistant criteria.
- **Evaluations relating to attack resistance:** Hash algorithms will be evaluated against attacks or observations that may threaten existing or proposed applications, or demonstrate some fundamental flaw in the design, such as exhibiting nonrandom behavior and failing statistical tests.
- **Other consideration factors:** The quality of the security arguments/proofs, the clarity of the documentation of the algorithm, the quality of the analysis on the algorithm performed by the submitters, the simplicity of the algorithm, and the confidence of NIST and the cryptographic community in the algorithm's long-term security may all be considered.

COST

- **Computational efficiency:** Computational efficiency refers to the execution speed of the algorithm. The evaluation of the computational efficiency of the candidate algorithms will be applicable to both hardware and software implementations. The Round 1 analysis by NIST will focus primarily on software implementations; hardware implementations will be addressed more thoroughly during the Round 2 analysis.
- **Memory requirements:** Memory requirements include such factors as gate counts for hardware implementations, and code size and RAM requirements for software implementations. The memory required to implement a candidate algorithm—for both hardware and software implementations of the algorithm—will be considered during the evaluation process. The Round 1 analysis will focus primarily on software implementations; hardware implementations will be addressed more thoroughly during Round 2.

Table V.1 NIST Evaluation Criteria for SHA-3 (page 2 of 2)

ALGORITHM AND IMPLEMENTATION CHARACTERISTICS

- **Flexibility:** Candidate algorithms with greater flexibility will meet the needs of more users than less flexible algorithms, and therefore, are preferable. However, some extremes of functionality are of little practical use (e.g., extremely short message digest lengths)—for those cases, preference will not be given. Some examples of ‘‘flexibility’’ may include (but are not limited to) the following:

  a. The algorithm has a tunable parameter, which allows the selection of a range of possible security/performance tradeoffs.

  b. The algorithm can be implemented securely and efficiently on a wide variety of platforms, including constrained environments, such as smart cards.

  c. Implementations of the algorithm can be parallelized to achieve higher performance efficiency.
- **Simplicity:** A candidate algorithm shall be judged according to relative simplicity of design.