Microsoft Windows SHA-3 Comments

As a major vendor of cryptographic implementations (in Microsoft Windows) and of software that uses cryptography, we appreciate this opportunity to comment on the selection criteria for SHA-3. We do not express an opinion about the security of any candidate, but instead comment on the non-security aspects of the selection that particularly affect the vendor & standards communities.

Full disclosure: a Microsoft employee, Niels Ferguson, is one of the designers of the Skein hash function, and also contributed to the comments below.

Adoption Criteria

SHA-256 and SHA-512 are already widely available and have not shown security weaknesses. To be widely adopted, SHA-3 has to provide benefits that outweigh the cost of adoption.

There is no precedent for a not-completely-broken crypto algorithm being successfully displaced by a new one without some significant other benefit. When the AES selection was announced, the industry switch to AES was driven by the enormous performance advantage (x2-x4) that AES offered over 3DES in a variety of settings.

Crypto algorithm adoption costs are significant, and require updating the infrastructure and consuming applications, including: data formats, protocols, certificate infrastructure, and applications with hardcoded assumptions around algorithm usage. All this imposes a significant opportunity cost on vendors and standards organizations, both in initial development and in subsequent interoperability and integration work. Vendors are hesitant to undertake such work in the absence of significant end-user benefits.

Reducing Implementation Complexity

Algorithms with multiple variants and/or options are costly – in practice the industry tends to converge on a single most widely deployed set of parameters to minimize the potential for interoperability issues. At the same time, retrofitting new hash values into old message formats sometimes requires truncation. We believe that the direction taken by NIST with SHA-512/t in FIPS 180-4, where a single basic algorithm can be safely truncated to any desired length, is a good way of addressing such issues.
We also recognize the benefits that tweakable parameters give in terms of adaptive abilities over time. Ultimately this comes down to a balance between the agility of an algorithm and the initial implementation costs.

The Importance of Performance

Microsoft products commonly use hash functions for wire protocols (IPsec/TLS), code signing, document/email signing, single sign-on (SSO) & user authentication, and other secondary uses such as key derivation. The most critical hash related performance bottlenecks are hit when verifying data integrity at rest and performing data authentication in wire protocols.

Performance is important in a number of ways, including:

1. Performance on general-purpose commodity processor architectures, such as those widely deployed in devices ranging from mobile hardware (smartphones and tablets) to embedded systems to desktop PCs to servers.
2. Efficiency on mobile form factor architectures, due to the direct impact on battery life, which is the primary design constraint.
3. Parallelizability due to the increase in parallel architectures (e.g., SIMD instruction sets) as a result of clock speed stabilization.
4. Hash algorithm versatility – usage of hash functions vary from HMAC-based algorithms such as Key Derivation Functions (i.e. very small message size) to network protocols (intermediate message size, about 1-10KB) to document and code signatures (large and potentially unlimited message sizes) and web-auth protocols. Good performance across all such scenarios is important.

To summarize, we believe that SHA-3 must exhibit great performance on as wide a range of commodity processors as possible, and should exploit the current Moore's law trend towards parallelism.

A very large number of applications use HMAC-SHA1 or HMAC-SHA256 for authenticating relatively short messages. Unfortunately, HMAC has a relatively high overhead, requiring two hash operations even for the shortest messages. A lower overhead MAC mode would potentially be beneficial to certain sets of applications, and may be an interesting addition to the capabilities provided by the SHA-2 family.

Conclusion

In conclusion, we commend NIST for taking on the challenge of standardizing SHA-3, and are grateful for the opportunity to comment. As a vendor and also as a participant in a large number of standards bodies, we believe that a SHA-3 selection which meets one or more of the above criteria will have the best chance of widespread and rapid adoption while also advancing the state of the art in cryptography.