

More engineering considerations for the SHA-3 hash function

Jean-Philippe Aumasson Dan Bernstein Charles Bouillaguet
Daniel Brown Orr Dunkelman Sebastiaan Indestege
Emilia Käsper Dmitry Khovratovich Jongsung Kim Özgül Küçük
Gaëtan Leurent Krystian Matusiewicz Florian Mendel
Ivica Nikolić Svetla Nikova Dag Arne Osvik Thomas Peyrin
Christian Rechberger Vincent Rijmen Ron Rivest
Martin Schläffer Søren Thomsen Elmar Tischhauser
Hirotaka Yoshida Dai Watanabe

“Table lookup: A huge security problem”

“Table lookup: A huge security problem”

- ▶ Side channel attacks only apply when a secret is involved (e.g. KDF, HMAC). Many applications do not hash secret information.
- ▶ There are other side channels, e.g., power. Modular additions are hard to protect against DPA.
- ▶ An implementation can be protected, at some cost, if required.

“Table lookup: A huge security problem”

“Table lookup: A huge security problem”

- ▶ Expect second AES competition in 2012 !

*“[AES-based designs are] insecure on CPUs
without AES instruction”*

*“[AES-based designs are] insecure on CPUs
without AES instruction”*

- ▶ Bitsliced implementations resist cache-timing attacks.
- ▶ They can actually be faster than table-based implementations.

*“Recommendation: avoid AES round
function”*

“Recommendation: avoid AES round function”

- ▶ Using AES also has advantages:
 - ▶ Security analysis
 - ▶ Confidence
 - ▶ Implementation
- ▶ ARX-based vs. AES-based?
 - ▶ No indication that one is better than the other.

*“Recommendation: optimize for 64-bit
(rather than 32-bit) performance”*

*“Recommendation: optimize for 64-bit
(rather than 32-bit) performance”*

- ▶ “The low end does not go away” (Bruce Schneier)
- ▶ Extreme optimisations for one platform (Intel Core2) often hurts other platforms.
- ▶ 32-bit optimised primitives are still fast on 64-bit, but not the other way around.

“Recommendation: also evaluate implementation without XMM registers”

“Recommendation: also evaluate implementation without XMM registers”

- ▶ Also pay attention to performance on 32-bit and 8-bit machines.

“Cannot use multiple cores”

“Cannot use multiple cores”

- ▶ Applications that cannot use multiple cores typically process only small messages.
- ▶ Some applications *can* use multiple cores, and those sometimes hash very long messages.

*“SHA-256 (20 c/B) is a performance
problem”*

“SHA-256 (20 c/B) is a performance problem”

- ▶ Why is it a problem?
 - ▶ Signatures? No.
 - ▶ HMAC? No, use fast dedicated MAC.
 - ▶ ...?
- ▶ “The security provided by an algorithm is the most important factor in the evaluation.” (NIST)

**And now for something
completely different...**

Sponges are bad

- ▶ Can't fit into small state after block.
- ▶ No key schedule to compute in parallel.
- ▶ No compression function; nothing reusable.
- ▶ Pseudo-collisions/preimages are easy to find.
- ▶ Large state \rightarrow slow full diffusion.
- ▶ Sponges are recent; not well studied.

Sponges are good

- ▶ Immediate use of block saves space.
- ▶ Very fast diffusion; extra speed.
- ▶ No counters.
- ▶ Not many sponges broken so far.
- ▶ Sponges are recent; they improve over other designs.

Disclaimer

- ▶ Most of us are involved with one or more SHA-3 candidates.
 - ▶ From 15 different teams in total.
- ▶ Every team has different priorities.
- ▶ Every design was made to fit those.
 - ▶ Not the other way around.

Jean-Philippe Aumasson Dan Bernstein Charles Bouillaguet
Daniel Brown Orr Dunkelman Sebastiaan Indestege Emilia Käsper
Dmitry Khovratovich Jongsung Kim Özgül Küçük Gaëtan Leurent
Krystian Matusiewicz Florian Mendel Ivica Nikolić Svetla Nikova
Dag Arne Osvik Thomas Peyrin Christian Rechberger
Vincent Rijmen Ron Rivest Martin Schläffer Søren Thomsen
Elmar Tischhauser Hirotaka Yoshida Dai Watanabe