CubeHash

D. J. Bernstein
University of Illinois at Chicago
CubeHash security is very well understood.

Third-party analyses by Aumasson, Brier, Dai, Ferguson, Khazaei, Khovratovich, Knellwolf, Lucks, McKay, Meier, Naya-Plasencia, Nikolic, Peyrin, Weinmann show that recommended CubeHash16/32–512 has a very solid security margin.

Thanks for all the analysis!
CubeHash16/32 has 768-bit pipe, so $\approx 2^{384}$ preimage security.

Alternate CubeHash16/1 option boosts pipe size and security, but quantum computers will limit SHA-3 to $2^{256}$ preimage security, so $2^{384}$ is already overkill.
CubeHash16/32 has 768-bit pipe, so $\approx 2^{384}$ preimage security.

Alternate CubeHash16/1 option boosts pipe size and security, but quantum computers will limit SHA-3 to $2^{256}$ preimage security, so $2^{384}$ is already overkill.

(Keccak speed advertisements have $\approx 2^{288}$ preimage security.)
CubeHash16/32 has 768-bit pipe, so $\approx 2^{384}$ preimage security.

Alternate CubeHash16/1 option boosts pipe size and security, but quantum computers will limit SHA-3 to $2^{256}$ preimage security, so $2^{384}$ is already overkill.

(Keccak speed advertisements have $\approx 2^{288}$ preimage security.)

CubeHash symmetries gain speed and are not a security problem.

CubeHash16/32 finalization: $\approx 320$ bytes, again overkill.
Those were the easy issues.

Harder issues, most interesting third-party analyses of CubeHash: increasingly sophisticated differential attacks.

Resulting collision costs: doable for CubeHash4/64; \(2^{71}\) estimate for CubeHash5/64; \(2^{132}\) estimate for CubeHash6/64; \(2^{180}\) estimate for CubeHash6/32.

Compared to CubeHash6/32, recommended CubeHash16/32 has \(> 2.5\times\) as many rounds.
Despite the security margin, CubeHash16/32–512 is about as fast as SHA-2. Slower on some old CPUs but faster on newer CPUs. 8.23 cycles/byte on Core i5 520. Will be < 5 cycles/byte on next year’s “AVX” Intel CPUs, thanks to 256-bit vectorization. Can even use future 512-bit AVX.

FPGA: Faster than SHA-256 in the same number of slices; and solidly beats SHA-512.

ASIC: Similar story.
We have other SHA-3 candidates with solid security margins and acceptable speed.

What’s special about CubeHash?
We have other SHA-3 candidates with solid security margins and acceptable speed.

What’s special about CubeHash?

CubeHash is the *smallest* high-security SHA-3 proposal.

Several meanings of “smallest”:

- Smallest memory use.
- Smallest description.
- Smallest code size.
- Smallest vector-code size.
- Smallest area in hardware.
We have other SHA-3 candidates with solid security margins and acceptable speed.

What’s special about CubeHash?

CubeHash is the *smallest* high-security SHA-3 proposal. Several meanings of “smallest”:

- Smallest memory use.
- Smallest description.
- Smallest code size.
- Smallest vector-code size.
- Smallest area in hardware.

(New: Mask bitsliced CubeHash \(\Rightarrow\) low-area DPA resistance.)
Bernet–Henzen–Kaeslin–Felber–Fichtner CubeHash8/1–512
ASIC: 7630 gate equivalents,
“particularly appealing for lightweight implementations.”

No cheating:
no “free external memory”;  
no “core functions only”; 
no security compromises.
Fast enough for almost all users.

No cheating:
no “free external memory”;
no “core functions only”; no security compromises.
Fast enough for almost all users.

Can anyone show me another SHA-3 candidate that fits full functionality into this area? ... with security above $2^{128}$?
How many users will care about performance of SHA-3?
Maybe 1/100 care about time.
Maybe 1/10 care about size.
CubeHash is the best choice whenever size is critical.
How many users will care about performance of SHA-3? Maybe 1/100 care about time. Maybe 1/10 care about size. CubeHash is the best choice whenever size is critical.

Some other proposals can fit into \( \approx 10000 \) gates if security is limited to \( 2^{128} \). The hardware cannot talk to high-security protocols that send 512-bit hashes. Implementation nightmare, as bad as having two SHA-3s.
Tiny ASIC takes advantage of tiny CubeHash state and tiny CubeHash code.

Same features help CubeHash on many other platforms.

Microcontroller? No problem.
Limited RAM size? No problem.
RAM competition? No problem.
ROM competition? No problem.

CubeHash fits anywhere.