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!

sh

ernstein

ty 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!

CubeHas
so $\approx 2^{38}$

Alternat
boosts p
but quar
SHA-3 t
so 2^{384} i

is 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 h
so $\approx 2^{384}$ preimag

Alternate CubeHa
boosts pipe size a
but quantum com
SHA-3 to 2^{256} pre
so 2^{384} is already

ago

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
so $\approx 2^{384}$ preimage security.

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

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.

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.

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

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.

(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.

sh security
vell understood.

arty analyses by
on, Brier, Dai,
n, Khazaei,
ovich, Knellwolf,
McKay, Meier,
asencia, Nikolic,
Weinmann
at recommended
sh16/32–512
ry solid security margin.

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.

(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 w

Harder i
third-par
increasin
different

Resulting
doable f
2^{71} esti
2^{132} esti
2^{180} esti

Compare
recomme
has $>$ 2.

y

tood.

es by

Dai,

,

ellwolf,

eier,

ikolic,

nended

512

curity margin.

e 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.

(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 ea

Harder issues, mos

third-party analyse

increasingly sophis

differential attacks

Resulting collision

doable for CubeHa

2^{71} estimate for C

2^{132} estimate for C

2^{180} estimate for C

Compared to Cube

recommended Cub

has $> 2.5\times$ as ma

rgin.

s!

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 interesti third-party analyses of Cube increasingly sophisticated differential attacks.

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

Compared to CubeHash6/32 recommended CubeHash16/ has $> 2.5\times$ as many rounds

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.

sh16/32 has 768-bit pipe,
34 preimage security.

e CubeHash16/1 option
pipe size and security,
ntum computers will limit
o 2^{256} preimage security,
is already overkill.

speed advertisements
2^{288} preimage security.)

sh symmetries gain speed
not a security problem.

sh16/32 finalization:
ytes, 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
CubeHas
is about

Slower o
but faste
8.23 cyc
Will be
on next
thanks t
Can eve

FPGA: F
in the sa
and solid

ASIC: Si

as 768-bit pipe,
ge security.

sh16/1 option
nd security,
puters will limit
eimage security,
overkill.

vertisements
age security.)

tries gain speed
urity problem.

nalization:
n 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 securi
CubeHash16/32–5
is about as fast as

Slower on some ol
but faster on newe
8.23 cycles/byte o
Will be < 5 cycles
on next year's "AV
thanks to 256-bit
Can even use futu

FPGA: Faster thar
in the same numb
and solidly beats S

ASIC: Similar stor

pipe,

tion
y,
limit
urity,

ts
ty.)

speed
em.

f

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
Will be < 5 cycles/byte
on next year's "AVX" Intel C
thanks to 256-bit vectorizati
Can even use future 512-bit

FPGA: Faster than SHA-25
in the same number of slices
and solidly beats SHA-512.

ASIC: Similar story.

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.

were the easy issues.

ssues, most interesting
rty analyses of CubeHash:
ngly sophisticated
ial attacks.

g collision costs:
or CubeHash4/64;
mate for CubeHash5/64;
mate for CubeHash6/64;
mate for CubeHash6/32.

ed to CubeHash6/32,
ended CubeHash16/32
.5× 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
with soli
and acce

What's s

sy issues.

st interesting
es of CubeHash:
sticated
s.

costs:
ash4/64;
ubeHash5/64;
CubeHash6/64;
CubeHash6/32.

eHash6/32,
beHash16/32
ny 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 SH
with solid security
and acceptable spe

What's special abo

ng
Hash:

/64;
6/64;
6/32.

2,
/32
.

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 candi
with solid security margins
and acceptable speed.

What's special about CubeH

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?

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?

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.

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?

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.)

the security margin,
sh16/32–512
as fast as SHA-2.

on some old CPUs
er on newer CPUs.
les/byte on Core i5 520.
$<$ 5 cycles/byte
year's "AVX" Intel CPUs,
o 256-bit vectorization.
n use future 512-bit AVX.

Faster than SHA-256
ame number of slices;
dly beats SHA-512.

imilar story.

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–
Fichtner
ASIC: 7
"particu
lightwei

No chea
no "free
no "core
no secur
Fast enc

ty margin,
512
SHA-2.

d CPUs
er CPUs.
n Core i5 520.
/byte
VX" Intel CPUs,
vectorization.
re 512-bit AVX.

n SHA-256
er of slices;
SHA-512.

y.

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–K
Fichtner CubeHas
ASIC: 7630 gate e
"particularly appea
lightweight implem

No cheating:
no "free external r
no "core functions
no security compro
Fast enough for al

520.

CPUs,
ion.
AVX.

6
s;

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
⇒ low-area DPA resistance.)

Bernet–Henzen–Kaeslin–Fell
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 u

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.

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
⇒ 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.

Can anyone show me another SHA-3 candidate that fits full functionality into this area?
... with security above 2^{128}?

e other SHA-3 candidates
d security margins
eptable speed.

special about CubeHash?

sh is the *smallest*
urity SHA-3 proposal.
meanings of "smallest":
est memory use.
est description.
est code size.
est vector-code size.
est area in hardware.

Mask bitsliced CubeHash
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.

Can anyone show me another
SHA-3 candidate that fits full
functionality into this area?
... with security above 2^{128}?

How ma
about pe

Maybe 1
Maybe 1
CubeHas
wheneve

A-3 candidates
margins
eed.

out CubeHash?

mallest
-3 proposal.
of "smallest":
y use.
tion.
ze.
code size.
hardware.

ced CubeHash
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.

Can anyone show me another SHA-3 candidate that fits full functionality into this area?
. . . with security above 2^{128}?

How many users w
about performance

Maybe 1/100 care
Maybe 1/10 care a
CubeHash is the b
whenever size is cr

dates

Hash?

al.
st":

Hash
)

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.

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 tin
Maybe 1/10 care about size
CubeHash is the best choice
whenever size is critical.

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.

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.

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.

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.

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.

Henzen–Kaeslin–Felber–
CubeHash8/1–512
630 gate equivalents,
larly appealing for
ght implementations."

ting:
external memory";
e functions only";
rity compromises.
ough for almost all users.

one show me another
candidate that fits full
ality into this area?
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.

Some other proposals can fit into ≈ 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 AS
tiny Cub
tiny Cub

Same fe
on many

Microco
Limited
Limited
RAM co
ROM co

CubeHa

aeslin–Felber–
h8/1–512
equivalents,
aling for
nentations."

memory";
s only";
omises.
most all users.

me another
that fits full
this area?
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.

Some other proposals can fit into ≈ 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 a
tiny CubeHash sta
tiny CubeHash coc

Same features help
on many other pla

Microcontroller? N
Limited RAM size
Limited ROM size
RAM competition
ROM competition

CubeHash fits ar

per–

s,

."

sers.

er

ull

?

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 ≈ 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 tiny CubeHash state *and* tiny CubeHash code.

Same features help CubeHas on many other platforms.

Microcontroller? No probler
Limited RAM size? No prob
Limited ROM size? No prob
RAM competition? No prob
ROM competition? No prob

CubeHash fits anywhere.

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

CubeHash fits anywhere.