SHA-3: should we care?

Jean-Philippe Aumasson

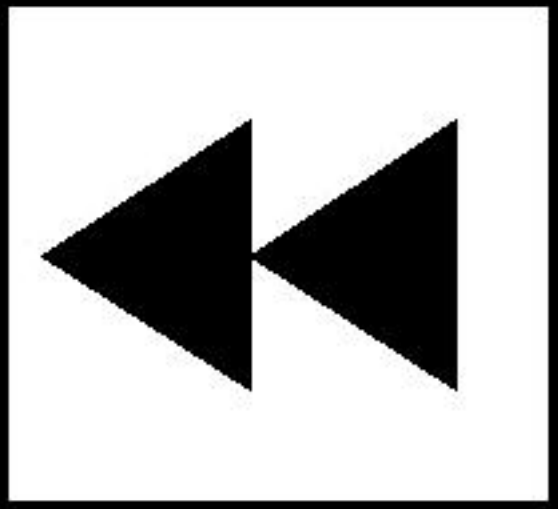
keccak.noekeon.org/sha3decision.html

NIST selects KECCAK for SHA-3

Definition for **kecak**:

Web definitions: Kecak (pronounced , alternate spellings: Ketjak and Ketjack) is a form of Balinese music drama, originated in the 1930s and is performed...
en.wikipedia.org/wiki/Kecak

2005



Collisions for Hash Functions

MD4, MD5, HAVAL-128 and RIPEMD

Xiaoyun Wang[1], Dengguo Feng[2], Xuejia Lai[3], Hongbo Yu[1]

The School of Mathematics and System Science, Shandong University, Jinan250100, China[1]

Institute of Software, Chinese Academy of Sciences, Beijing100080, China[2]

Dept. of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China[3]

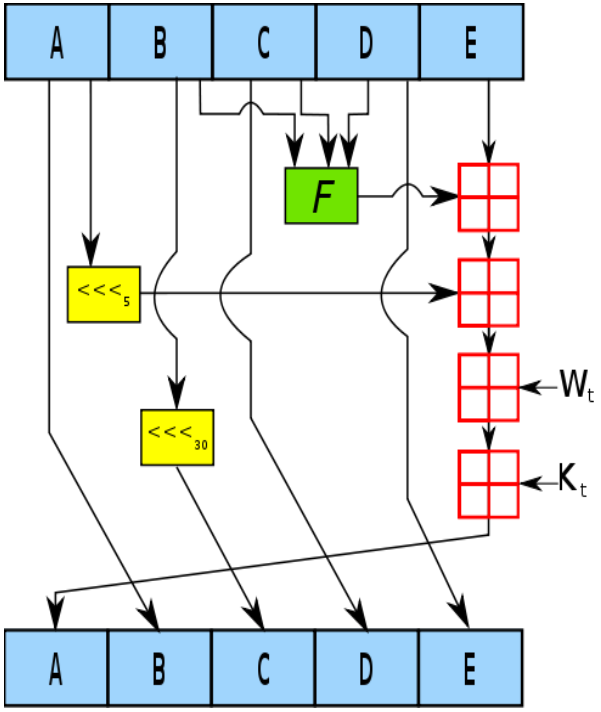
xywang@sdu.edu.cn[1]

revised on August 17, 2004

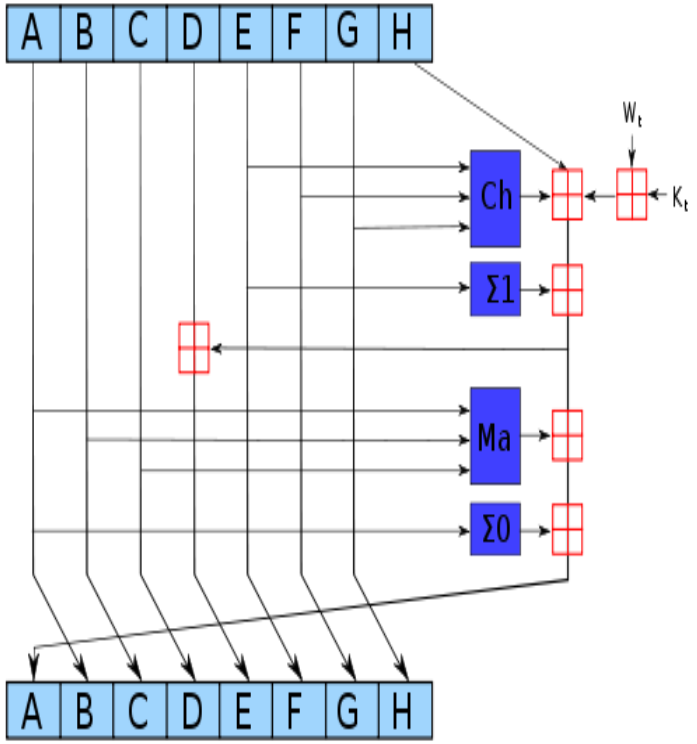
Finding Collisions in the Full SHA-1

Xiaoyun Wang[1⋆], Yiqun Lisa Yin[2], and Hongbo Yu[3]

[1] Shandong University, Jinan 250100, China, xywang@sdu.edu.cn
[2] Independent Security Consultant, Greenwich CT, US, yyin@princeton.edu
[3] Shandong University, Jinan250100, China, yhb@mail.sdu.edu.cn



SHA-1



SHA-2


EVERYBODY PANIC!!!
O NOES!
OMG!

Nov 2007

DEPARTMENT OF COMMERCE

National Institute of Standards and Technology

[Docket No.: 070911510–7512–01]

Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA–3) Family

AGENCY: National Institute of Standards and Technology, Commerce.

ACTION: Notice and request for nominations for candidate hash algorithms.

Nov 2007

NIST has decided that it is prudent to develop a new hash algorithm to augment and revise FIPS 180–2. The new hash algorithm will be referred to as ''SHA–3'', and will be developed through a public competition, much like the development of the Advanced Encryption Standard (AES). NIST intends that SHA–3 will specify an unclassified, publicly disclosed algorithm(s), which is available worldwide without royalties or other intellectual property restrictions, and is capable of protecting sensitive information for decades. Following the

Nov 2007

NIST expects SHA–3 to have a security strength that is at least as good as the hash algorithms currently specified in FIPS 180–2, and that this security strength will be achieved with significantly improved efficiency. NIST

Nov 2008

64 submissions received

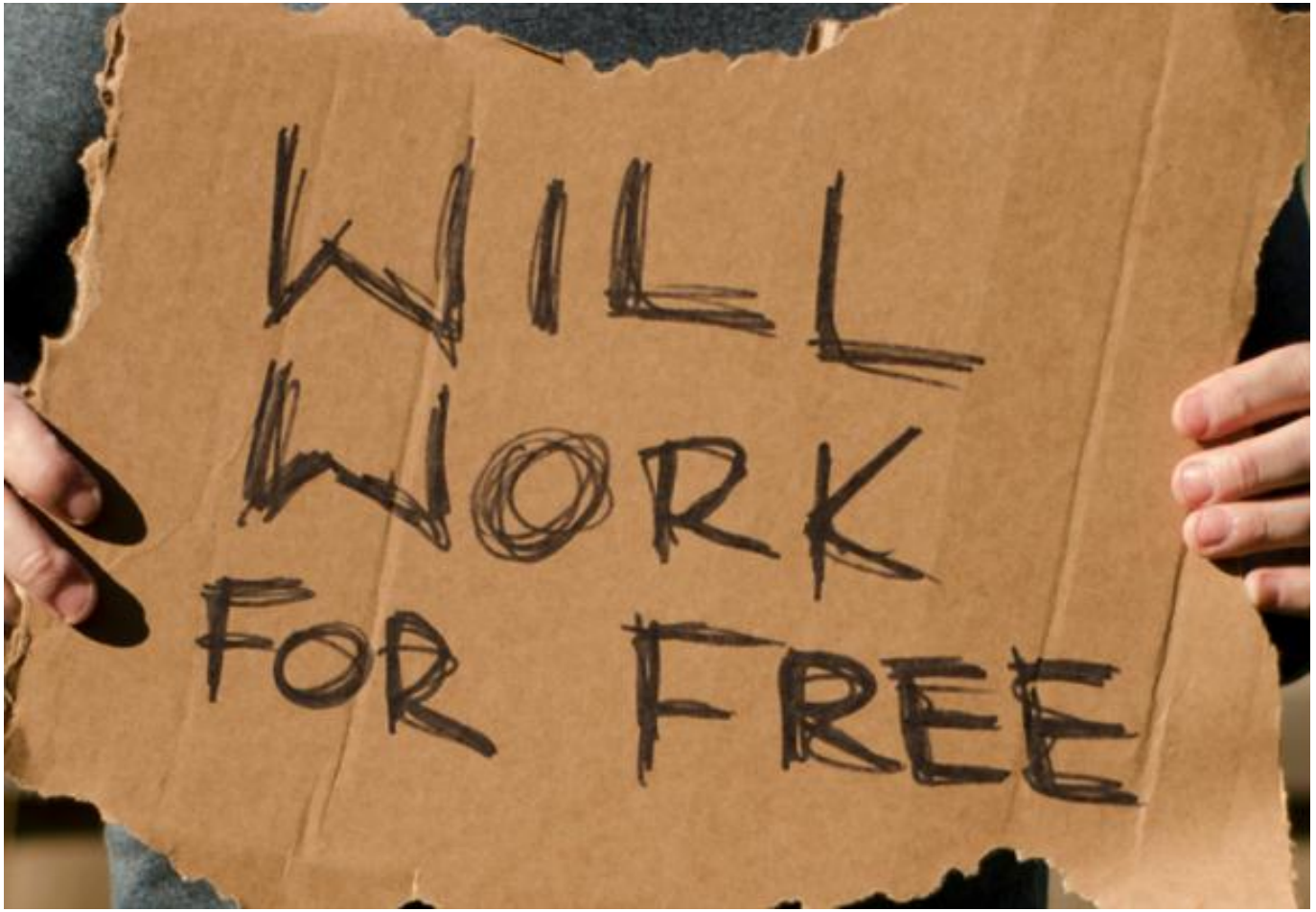
51 “complete and proper”

14 in the “second round”,

5 in the “final”


SONY
certicom™
IBM
Microsoft®
QUALCOMM®
BT
NXP
orange™
gemalto
intel®
HITACHI
Inspire the Next
EADS
KATHOLIEKE UNIVERSITEIT
LEUVEN
KOREA
UNIVERSITY
1905
EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Technion
Israel Institute of
Technology
Sandia
National
Laboratories
ETH
MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
1861
MENS ET MANUS
n|w
Fachhochschule
Nordwestschweiz
LETTRES
SCIENCES
ÉCOLE NORMALE SUPÉRIEURE
TU
Graz
INRIA


57
ON MY
WAY
31



Most security and performance evaluation by submitters and third parties, not NIST

Cryptanalysis

Hash Name	Principal Submitter	Status	Best Attack on Main NIST Requirements	Best Attack on other Hash Requirements
Abacus	Neil Sholer	in round 1	2nd-preimage	
ARIRANG	Jongin Lim	in round 1		
AURORA	Masahiro Fujita	in round 1	2nd preimage	
Blender	Colin Bradbury	in round 1	collision, preimage	near-collision
Boole	Greg Rose	in round 1	collision	
Cheetah	Dmitry Khovratovich	in round 1		length-extension
CHI	Phillip Hawkes	in round 1		
CRUNCH	Jacques Patarin	in round 1		length-extension
DCH	David A. Wilson	in round 1	collision	
Dynamic SHA	Xu Zijie	in round 1	collision	length-extension
Dynamic SHA2	Xu Zijie	in round 1	collision	length-extension
ECOH	Daniel R. L. Brown	in round 1	2nd preimage	
Edon-R	Danilo Gligoroski	in round 1	preimage	
EnRUPT	Sean O'Neil	in round 1	collision	
ESSENCE	Jason Worth Martin	in round 1	collision	
FSB	Matthieu Finiasz	in round 1		
HASH 2X	Jason Lee	not in round 1	2nd-preimage	
Khichidi-1	M. Vidyasagar	in round 1	collision	

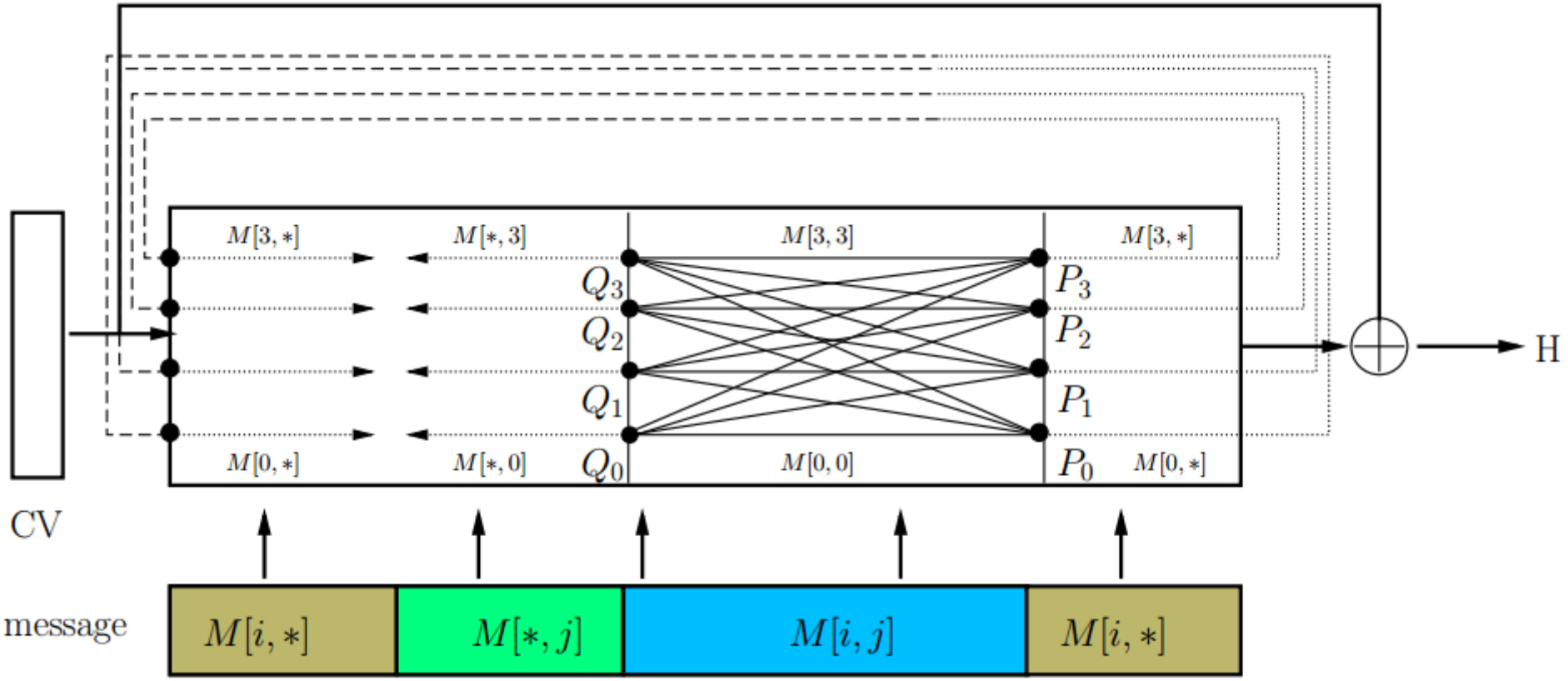
http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo

Cryptanalysis

for each b^1 in L_{B_1} and for $j = [1, 2]$ **do**
Find an element in $L^j_{\#8,b^1}$ such that $h_j^{-1}(y_j) \oplus h_j^{-1}(y_j \oplus b_j^1) = \Delta_j^{\#8}$ and store $(h_1^{-1}(y_1), \Delta_1^{\#8}, h_2^{-1}(y_2), \Delta_2^{\#8}, b^1)$ in L_{aux_1}.
for each b^2 in L_{B_2} and for $j = [5, 6]$ **do**
Find an element in $L^j_{\#8,b^2}$ such that $h_j^{-1}(y_j) \oplus h_j^{-1}(y_j \oplus b_j^2) = \Delta_j^{\#8}$ and store $(h_5^{-1}(y_5), \Delta_5^{\#8}, h_6^{-1}(y_6), \Delta_6^{\#8}, b^2)$ in L_{aux_2}.
for each $(h_1^{-1}(y_1), \Delta_1^{\#8}, h_2^{-1}(y_2), \Delta_2^{\#8}, b^1)$ in L_{aux_1} and **for** each $(h_5^{-1}(y_5), \Delta_5^{\#8}, h_6^{-1}(y_6), \Delta_6^{\#8}, b^2)$ in L_{aux_2} **do**
Compute $V_1' = h_1^{-1}(y_1) \oplus 3 \times h_5^{-1}(y_5)$ and $V_2' = h_2^{-1}(y_2) \oplus 3 \times h_6^{-1}(y_6)$, and store $((h_1^{-1}(y_1), \Delta_1^{\#8}, h_2^{-1}(y_2), \Delta_2^{\#8}, b^1), (h_5^{-1}(y_5), \Delta_5^{\#8}, h_6^{-1}(y_6), \Delta_6^{\#8}, b^2))$ in a hash table T indexed by these (V_1', V_2').
for Δ_Y from 0 to $2^{64} - 1$ **do**
Determine by BigMC $\Delta_{j'}^{\#8}$ for $j' = 3, 4, 7, 8$; and $\Delta_j^{\#6}$ for $j \in [1, \ldots, 12]$.
for i from 1 to 12 **do**
Find the element from $L^i_{\#6}$ such that $g_i(x_i) \oplus g_i(x_i, a_i) = \Delta_i^{\#6}$.
Compute with them by MC the values $d_i^{\#7}$ of the active diagonals in #7 and then $V_j = 2 \times d_j^{\#7} \oplus d_{j+4}^{\#7} \oplus d_{j+8}^{\#7} \oplus 9 \times d_j^{\#7} \oplus 3 \times d_{j+4}^{\#7} \oplus 6 \times d_{j+8}^{\#7}$ for $j = 1, 2$.
if there is an element such that $V_1' = V_1$ and $V_2' = V_2$ in T (*one on average, determines* b^1 *and* b^2) **then**
Find from $L^{j'}_{\#8,b^1}$ the element $(h_{j'}^{-1}(y_{j'}), \Delta_{j'}^{\#8})$ for $j' = 3, 4$. This determines y_3 and y_4.
Find from $L^{j'}_{\#8,b^2}$ the element $(h_{j'}^{-1}(y_{j'}), \Delta_{j'}^{\#8})$ for $j' = 7, 8$. This determines y_7 and y_8.

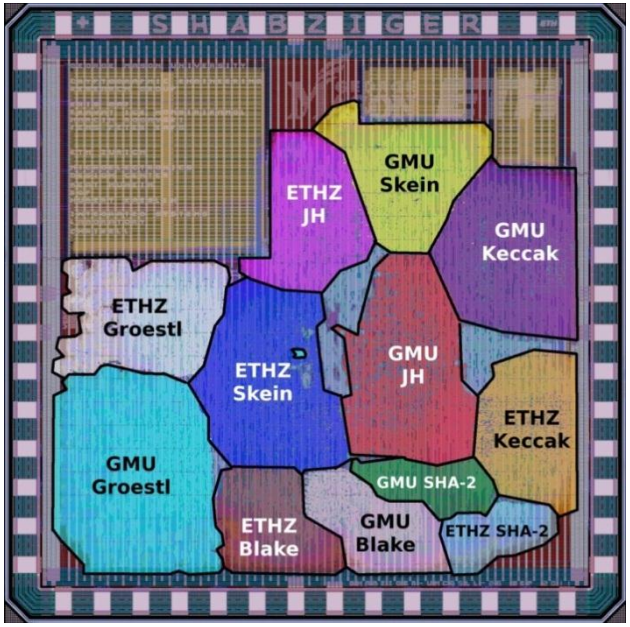
http://eprint.iacr.org/2010/607.pdf

Cryptanalysis

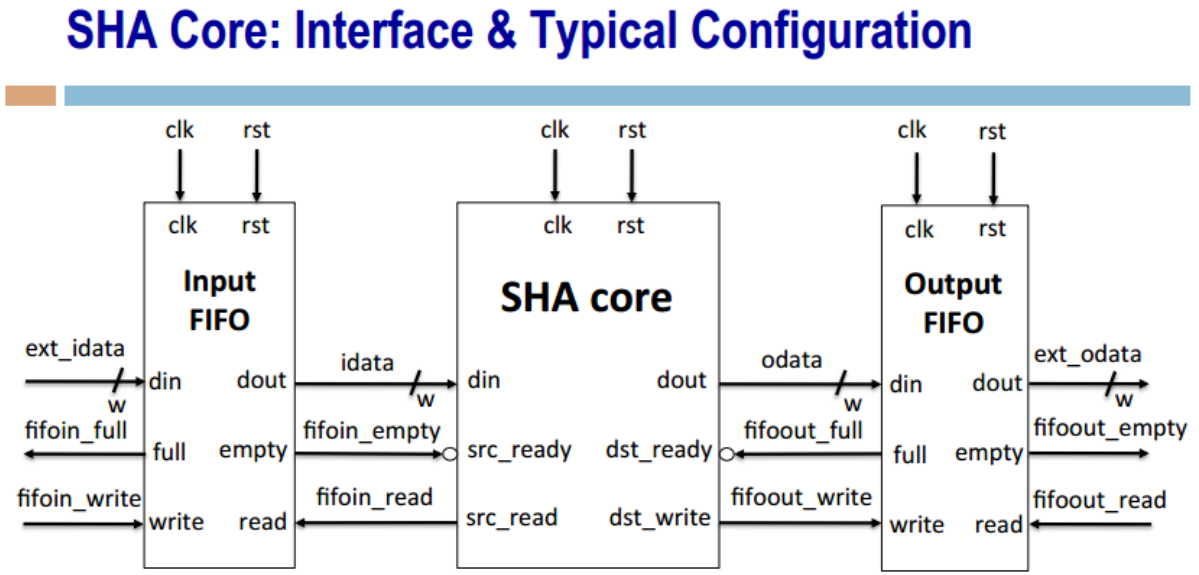


http://eprint.iacr.org/2011/286.pdf

Hardware performance evaluation (ASIC, FPGA)



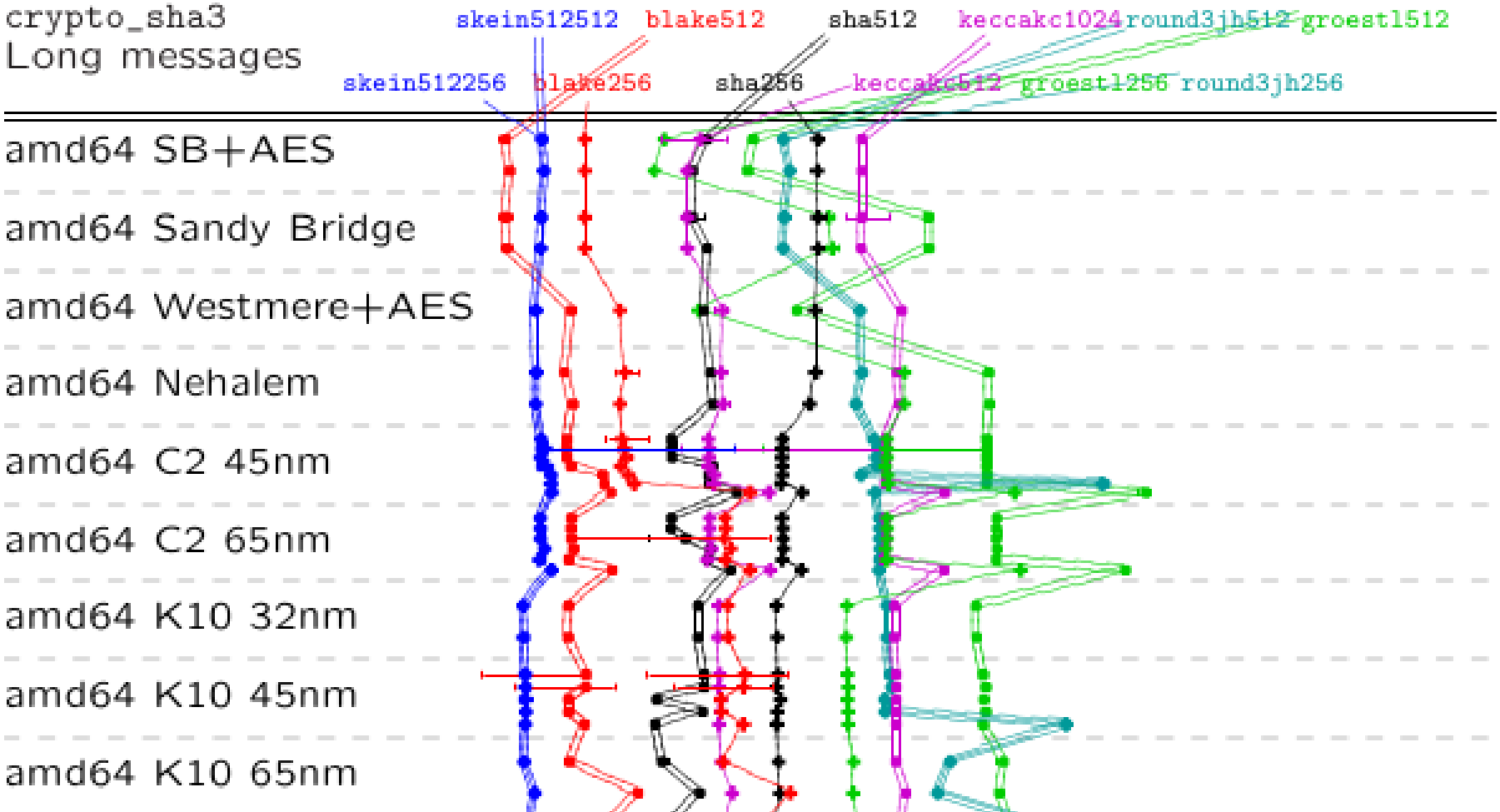
http://www.iis.ee.ethz.ch/~sha3/



http://ece.gmu.edu/~kgaj/publications/conferences/GMU_CHES_2010_slides.pdf

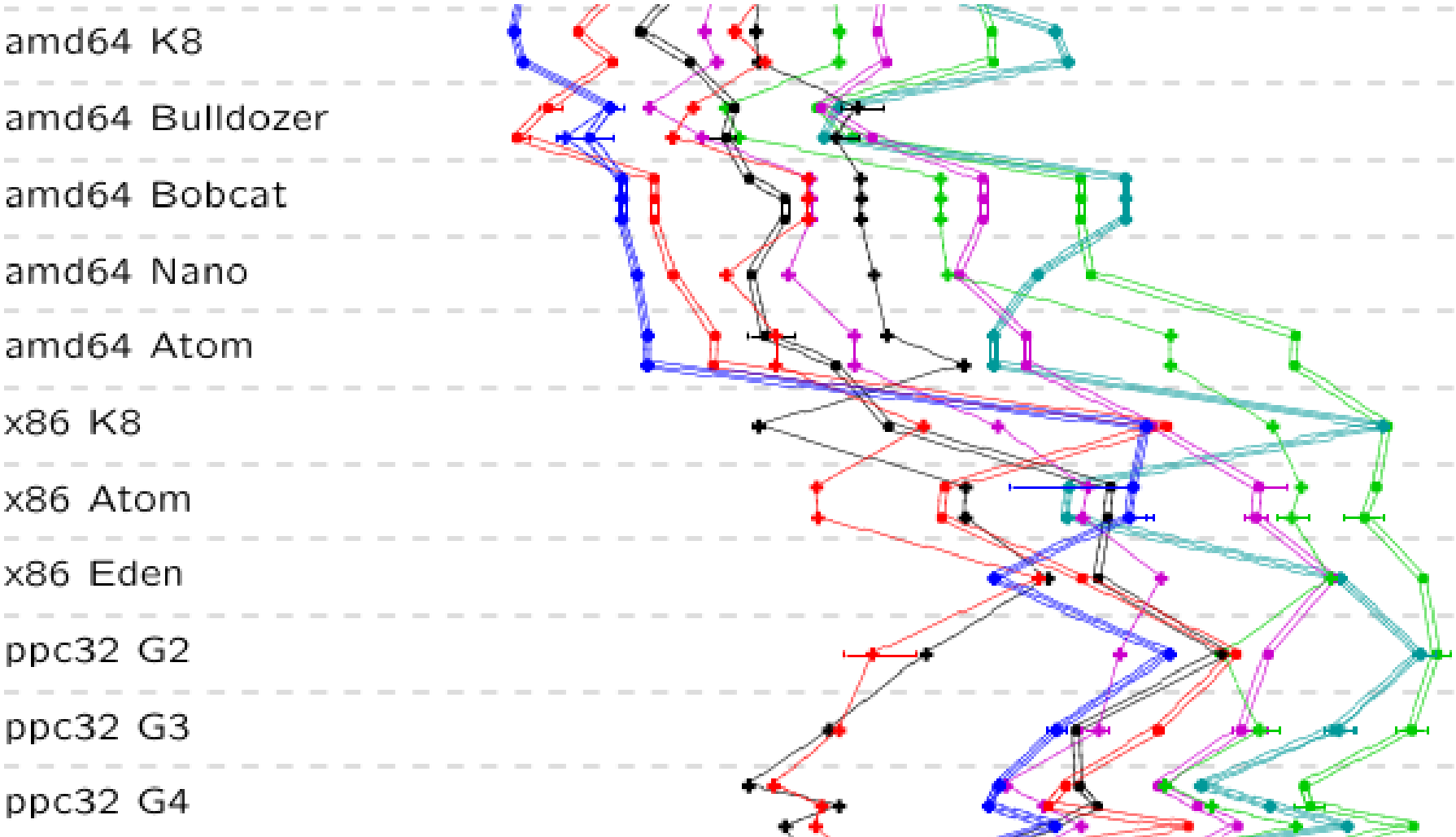
Software performance evaluation:

tools developed to find the fastest combination implementation/compiler/options for each CPU



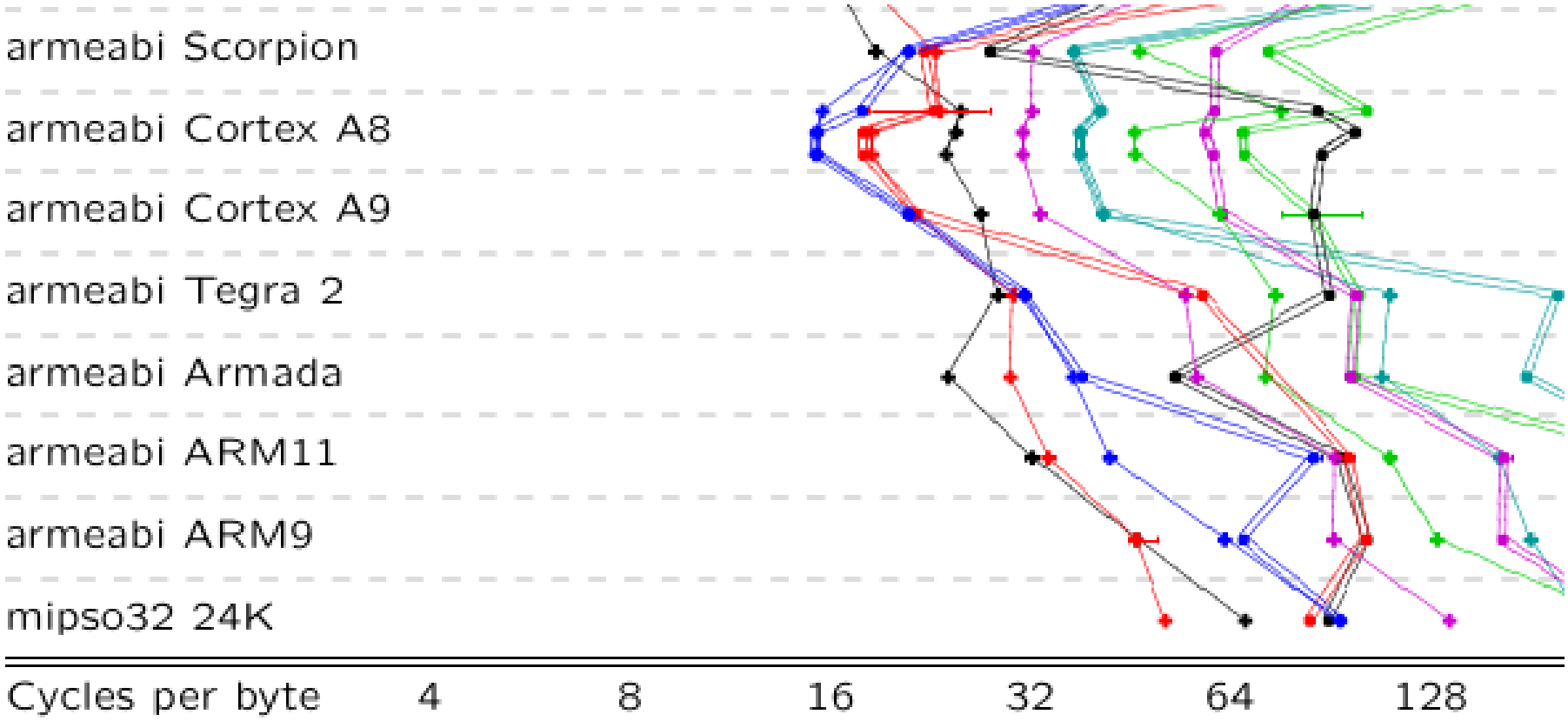
Software performance evaluation:

tools developed to find the fastest combination implementation/compiler/options for each CPU



Software performance evaluation:

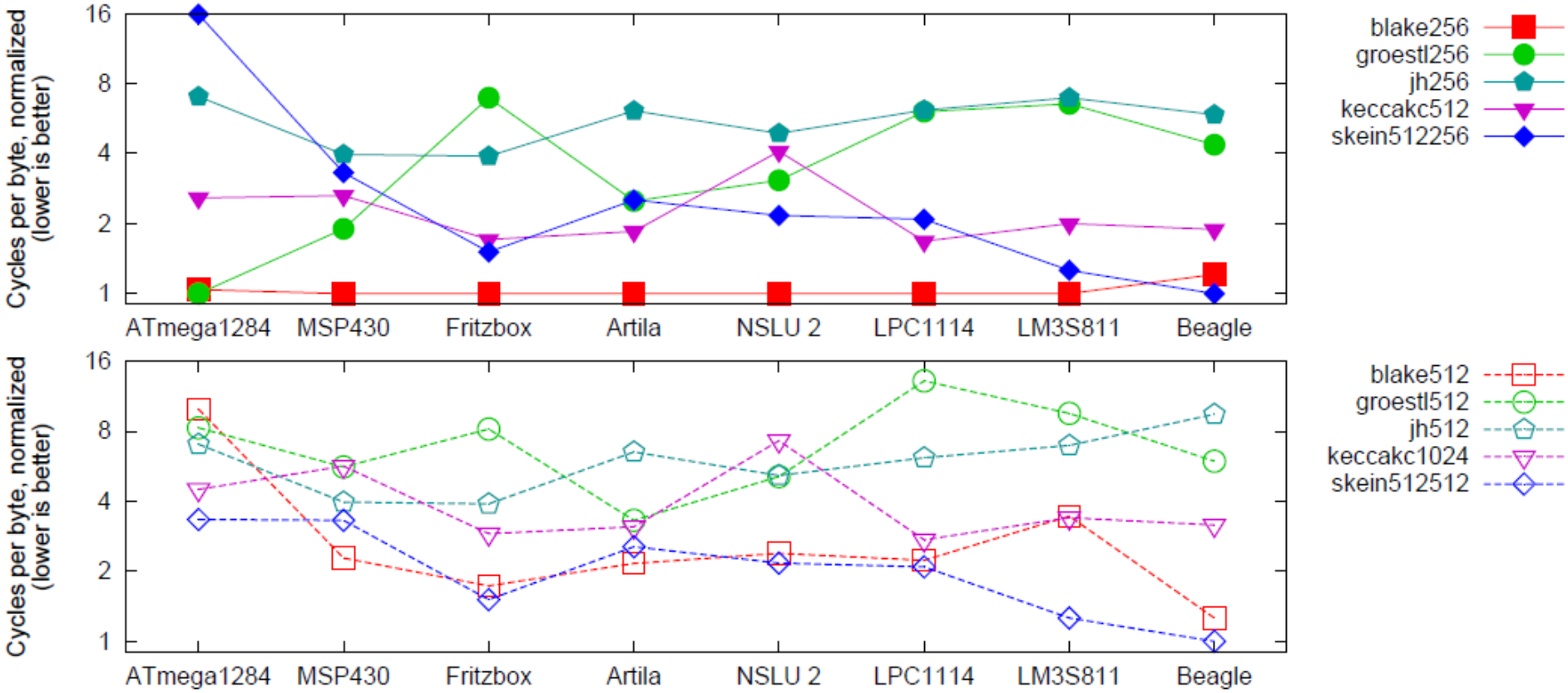
tools developed to find the fastest combination implementation/compiler/options for each CPU



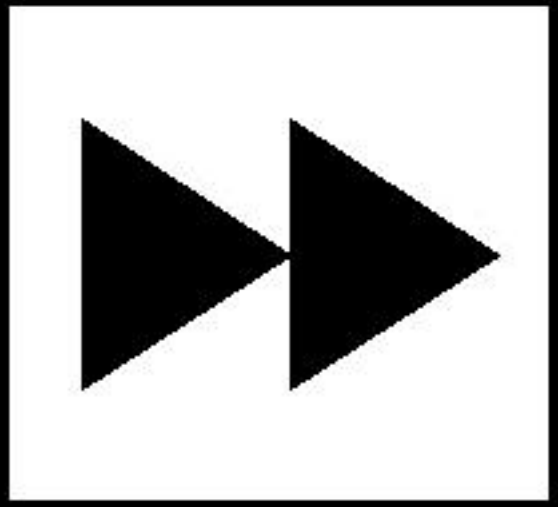
More on the eBACS project page: http://bench.cr.yp.to/

Software performance evaluation:

tools developed to find the fastest combination implementation/compiler/options for each CPU



More on the XBX project page: http://xbx.das-labor.org



2012

BLAKE

Groestl

JH

Keccak

Skein

BLAKE

Groestl

JH

Keccak

Skein

“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**.”

NIST PR

“Keccak ***complements*** the existing SHA-2 family of hash algorithms well.”

NIST email announcement

“An attack that could work on SHA-2 most likely would not work on Keccak because the two algorithms are designed so differently

…

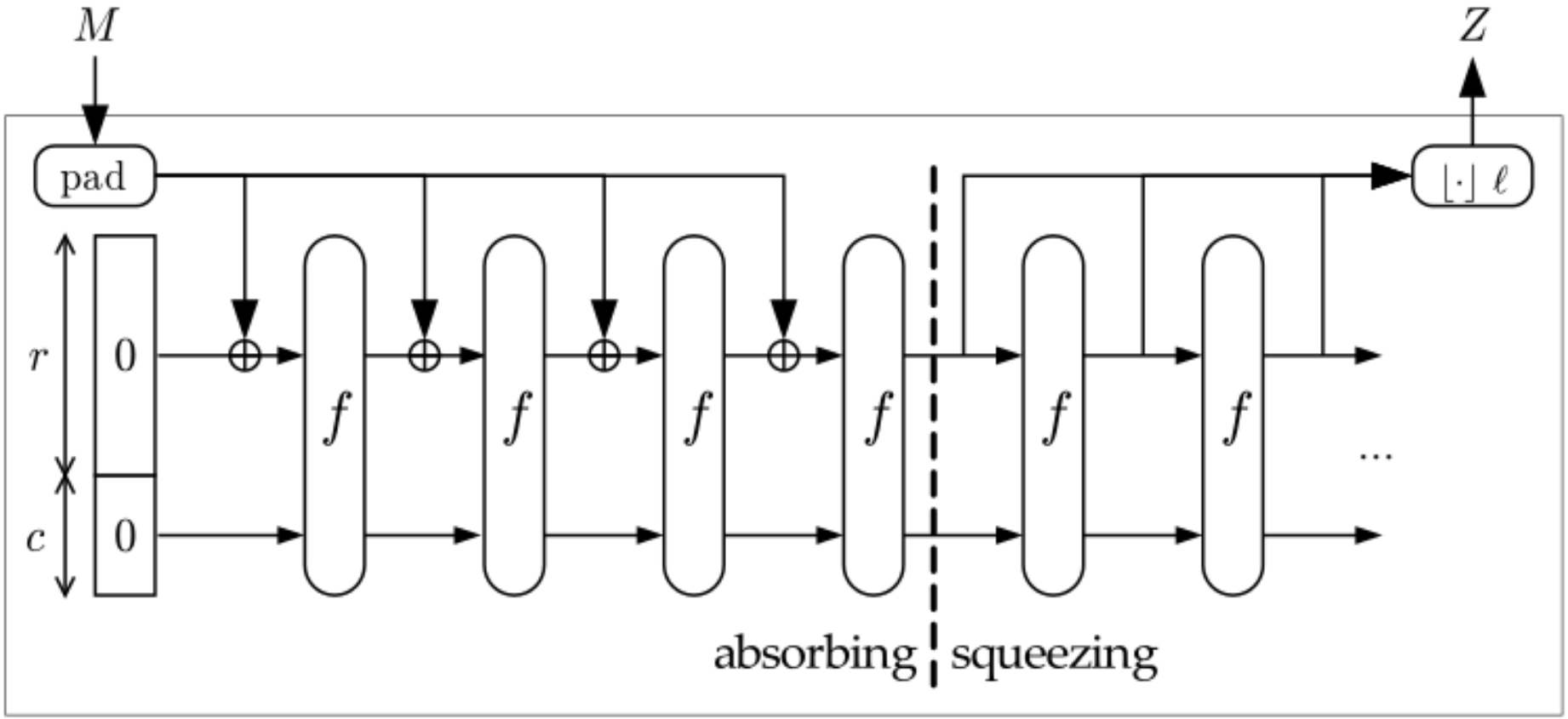
it immediately provides an **essential insurance** policy in case SHA-2 is ever broken”

NIST PR

The Keccak sponge function family

Guido Bertoni[1], Joan Daemen[1], Michaël Peeters[2] and Gilles Van Assche[1]

[1]STMicroelectronics
[2]NXP Semiconductors



sponge

Keccak's core permutation

$$\mathrm{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta, \text{ with}$$

$$\theta: \quad a[x][y][z] \leftarrow a[x][y][z] + \sum_{y'=0}^{4} a[x-1][y'][z] + \sum_{y'=0}^{4} a[x+1][y'][z-1],$$

$$\rho: \quad a[x][y][z] \leftarrow a[x][y][z-(t+1)(t+2)/2],$$

with t satisfying $0 \leq t < 24$ and $\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}^t \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$ in $\mathrm{GF}(5)^{2\times 2}$,

or $t = -1$ if $x = y = 0$,

$$\pi: \quad a[x][y] \leftarrow a[x'][y'], \text{ with } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix},$$

$$\chi: \quad a[x] \leftarrow a[x] + (a[x+1] + 1)a[x+2],$$

$$\iota: \quad a \leftarrow a + \mathrm{RC}[i_r].$$

http://keccak.noekeon.org/Keccak-reference-3.0.pdf

Actually simpler than it looks

```
Round[b](A,RC) {
  θ step
  C[x] = A[x,0] xor A[x,1] xor A[x,2] xor A[x,3] xor A[x,4],   forall x in 0…4
  D[x] = C[x-1] xor rot(C[x+1],1),                            forall x in 0…4
  A[x,y] = A[x,y] xor D[x],                                   forall (x,y) in (0…4,0…4)

  ρ and π steps
  B[y,2*x+3*y] = rot(A[x,y], r[x,y]),                         forall (x,y) in (0…4,0…4)

  χ step
  A[x,y] = B[x,y] xor ((not B[x+1,y]) and B[x+2,y]),          forall (x,y) in (0…4,0…4)

  ι step
  A[0,0] = A[0,0] xor RC

  return A
}
```

http://keccak.noekeon.org/specs_summary.html

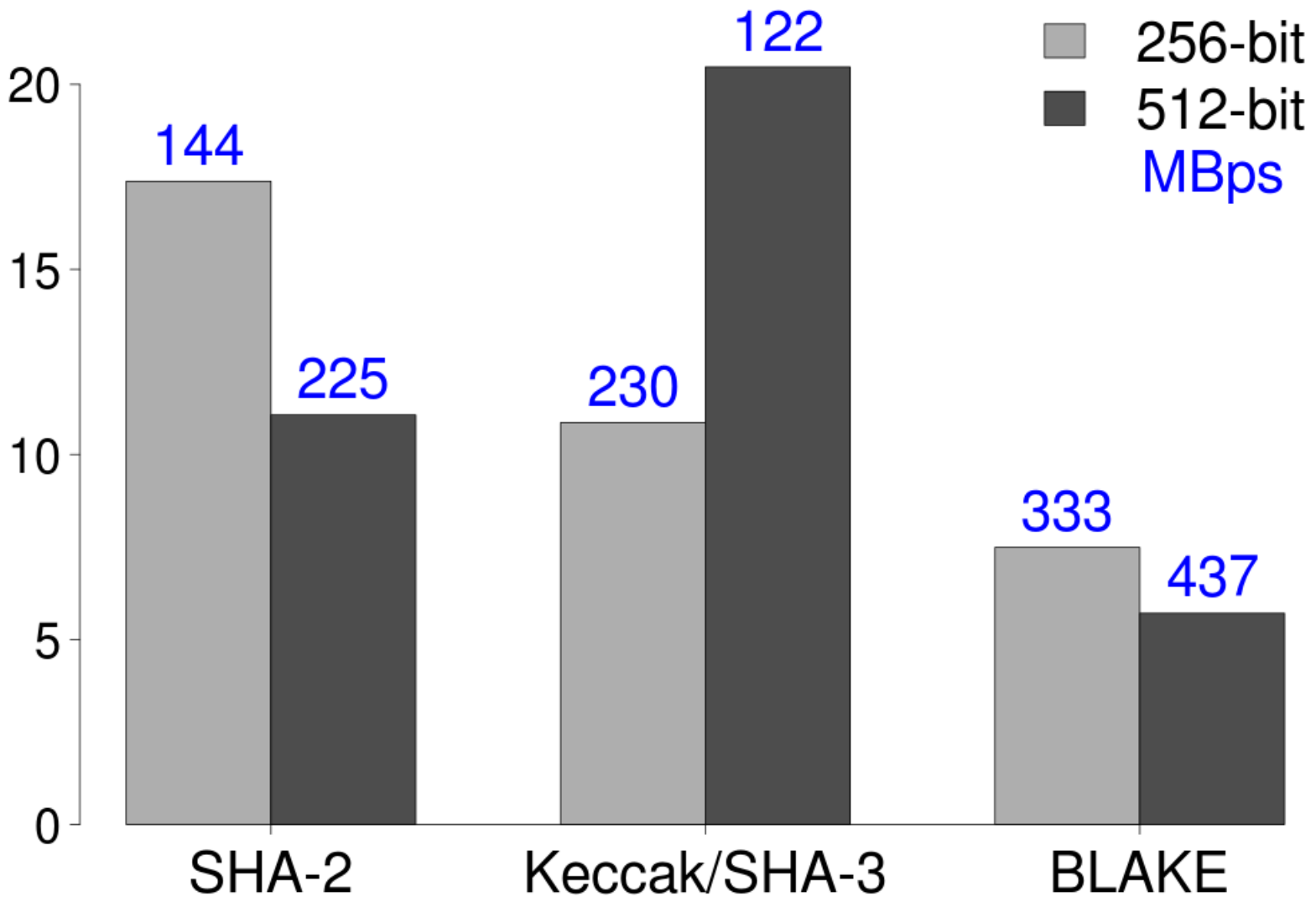
Many techniques for fast implementations

Algorithm 3 Plane-per-plane processing with bit interleaving of factor s

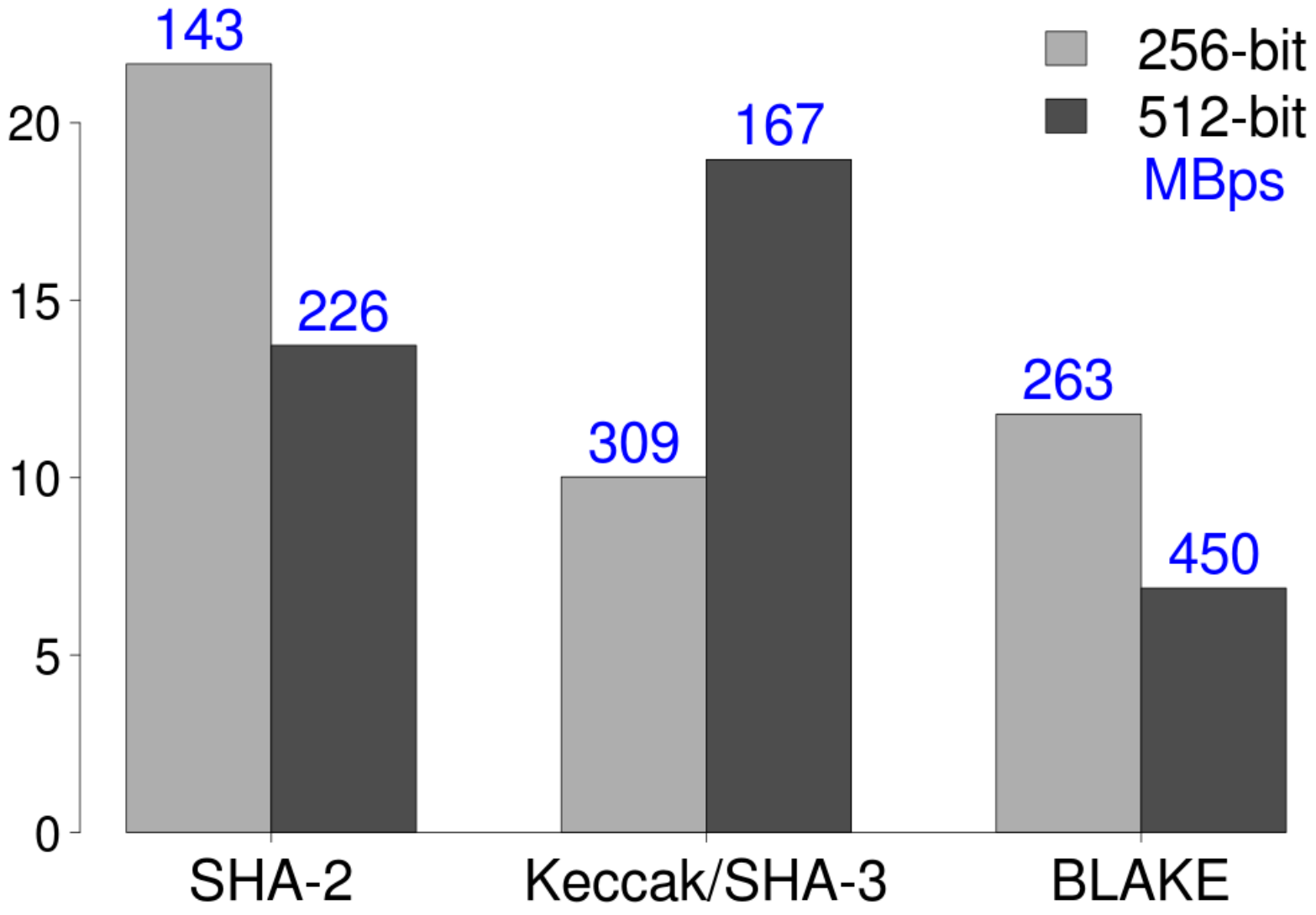
for $x = 0$ to 4 **and** $\zeta = 0$ to $s-1$ **do**
  $C[x,\zeta] = A[x,0,\zeta] \oplus A[x,1,\zeta] \oplus A[x,2,\zeta] \oplus A[x,3,\zeta] \oplus A[x,4,\zeta]$
end for
for $x = 0$ to 4 **do**
  $D[x,0] = C[x-1,0] \oplus \mathrm{ROT}(C[x+1,s-1],1)$
  for $\zeta = 1$ to $s-1$ **do**
    $D[x,\zeta] = C[x-1,\zeta] \oplus C[x+1,\zeta-1]$
  end for
end for
for $y = 0$ to 4 **and** $\zeta = 0$ to $s-1$ **do**
  for $x = 0$ to 4 **do**
    Let $(x',y')^{\mathrm{T}} = M^{-1}(x,y)^{\mathrm{T}}$ and $\zeta' = \zeta - r[x',y'] \mod s$
    Let $r = \lfloor \frac{r[x',y']}{s} \rfloor + 1$ if $\zeta < r[x',y'] \mod s$, or $r = \lfloor \frac{r[x',y']}{s} \rfloor$ otherwise
    $B[x] = \mathrm{ROT}((A[x',y',\zeta'] \oplus D[x',\zeta']),r)$
  end for
  for $x = 0$ to 4 **do**
    $E[x,y,\zeta] = B[x] \oplus ((\mathrm{NOT}\, B[x+1]) \,\mathrm{AND}\, B[x+2])$
  end for
end for
$E[0,0,\zeta] = E[0,0,\zeta] \oplus \mathrm{RC}_s[i,\zeta]$ **for** $\zeta = 0$ to $s-1$

http://keccak.noekeon.org/Keccak-implementation-3.2.pdf

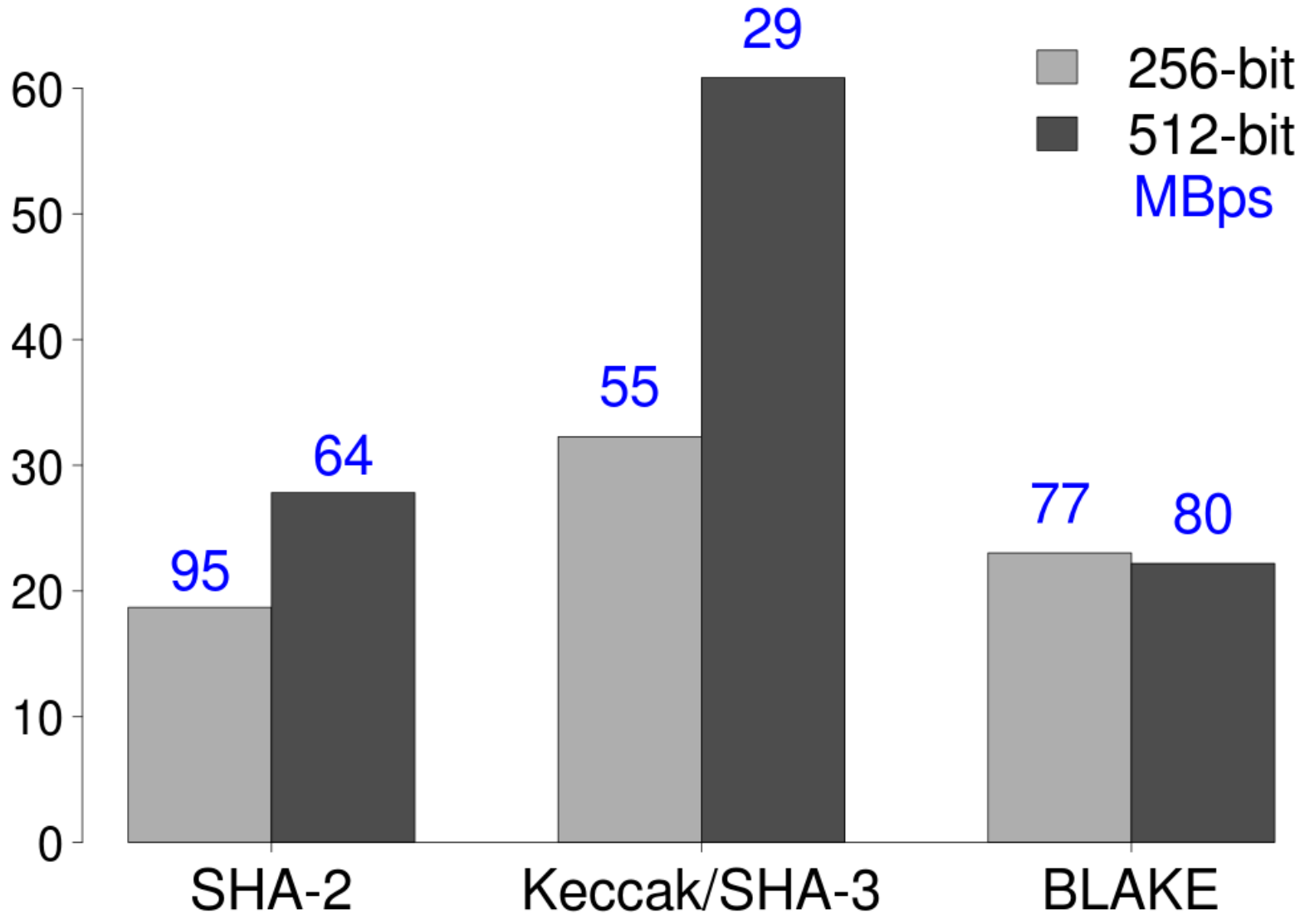
Intel Core i5-2400S (Sandy Bridge, 4 x 2495 MHz)



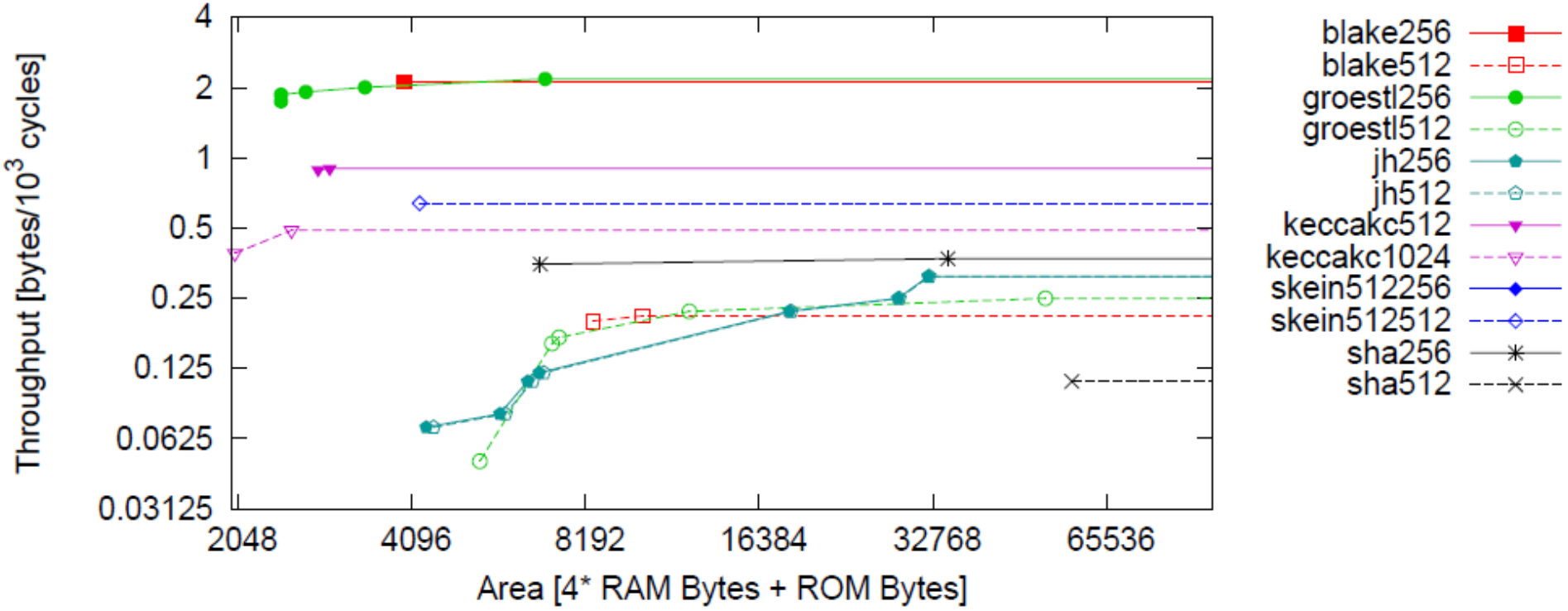
AMD FX-8120 (Bulldozer, 4 x 3100 MHz)



Qualcomm Snapdragon S3 APQ8060 (2 x 1782 MHz)

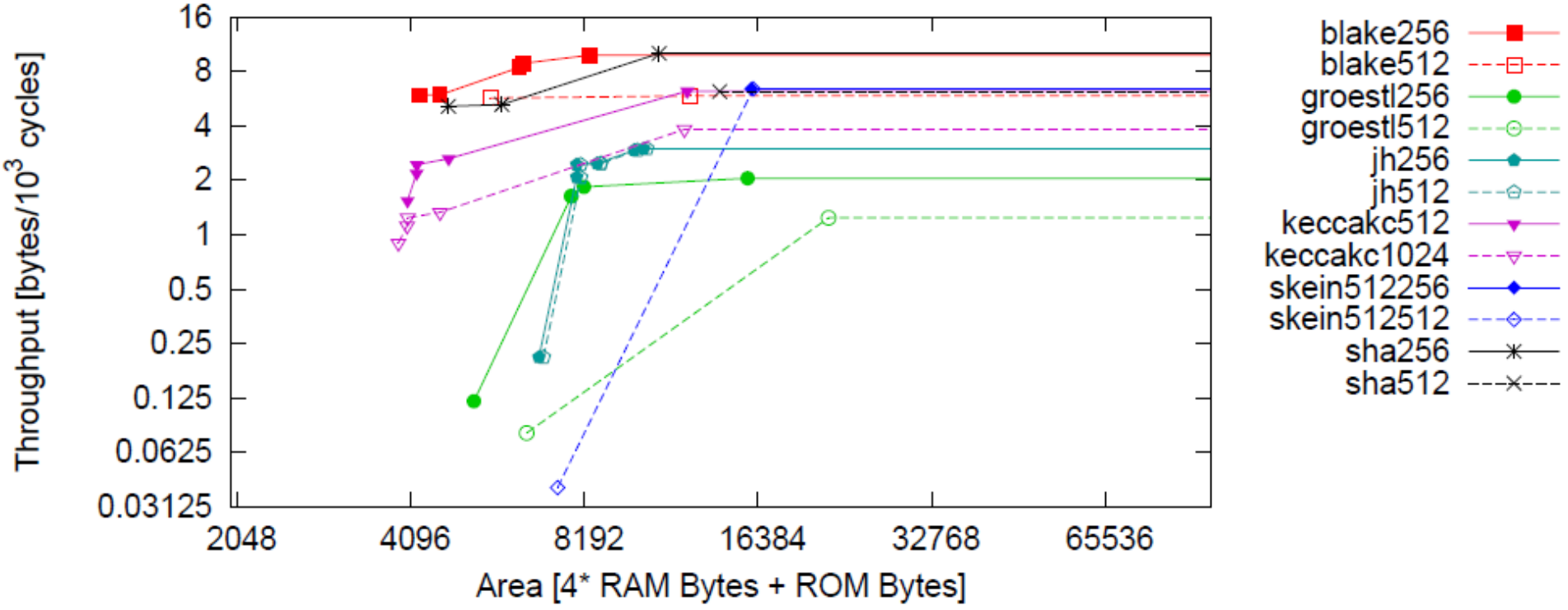


AVR (8-bit): Atmel ATmega1284P

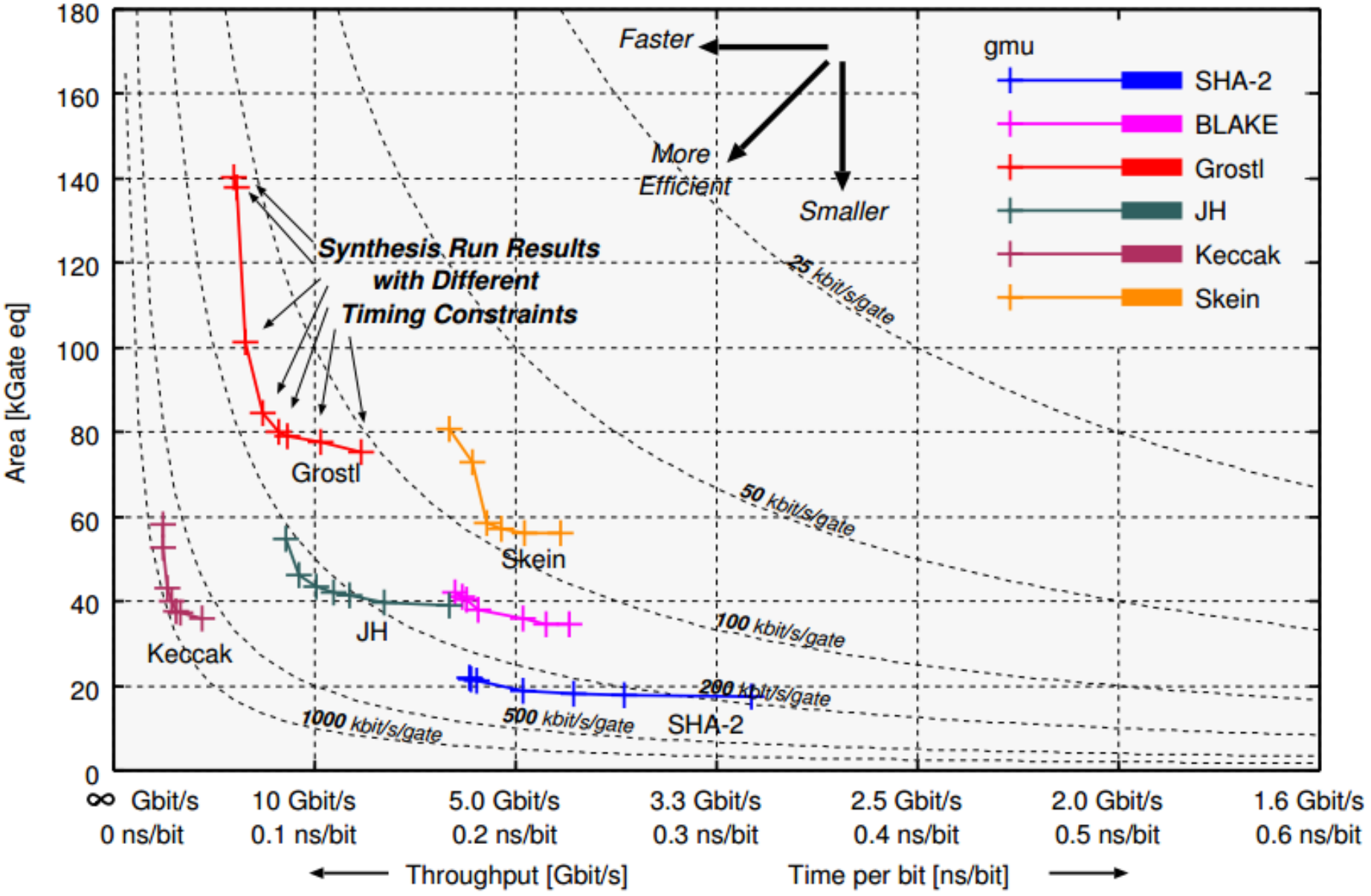


Credit: XBX project http://xbx.das-labor.org

MIPS (32-bit): Texas Instruments AR7

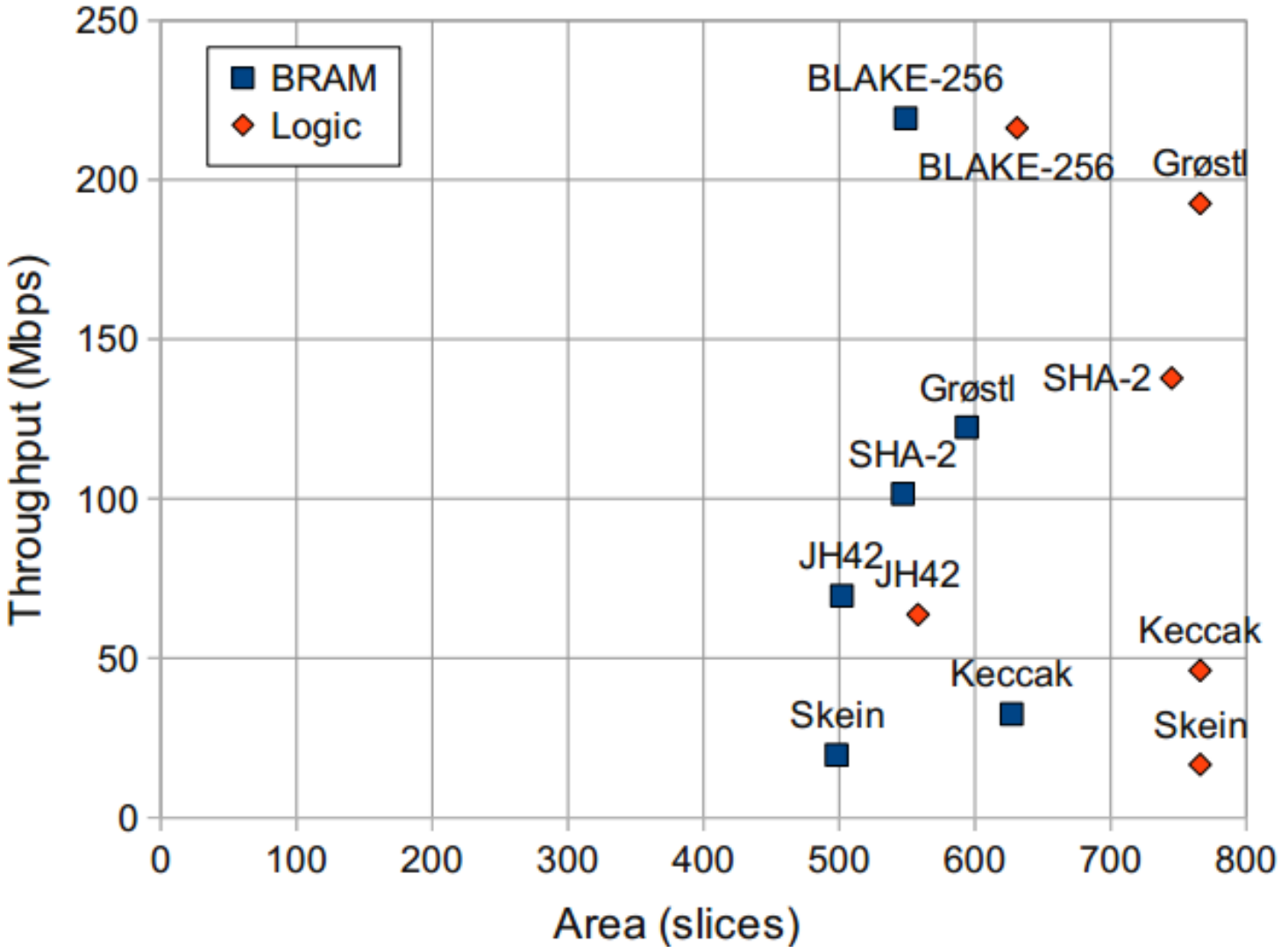


Credit: XBX project http://xbx.das-labor.org



http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/March2012/documents/presentations/GURKAYNAK_presentation.pdf

Throughput versus Area on Spartan-3



http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/March2012/documents/presentations/KAPS_presentation.pdf

Keccak's security: despite appealing prizes, very tough to attack…

keccak.noekeon.org/prize2_result.html

Congratulations to the winners of the second KECCAK cryptanalysis prize

9 September 2009

We are happy to announce that **Jean-Philippe Aumasson** and **Willi Meier** are the winners of the second KECCAK cryptanalysis prize for their note entitled *Zero-sum distinguishers for reduced KECCAK-f and for the core functions of Luffa and Hamsi*. The awarded Bialetti coffee machine and its full travel set were handed over to Jean-Philippe yesterday at the rump session of CHES 2009 in Lausanne. *Cong*

Beer-recovery attack

Jean-Philippe Aumasson Dmitry Khovratovich

Unaligned Rebound Attack: Application to Keccak

Alexandre Duc[1,*], Jian Guo[2,†], Thomas Peyrin[3,‡], and Lei Wei[3,§]

[1] Ecole Polytechnique Fédérale de Lausanne, Switzerland
[2] Institute for Infocomm Research, Singapore
[3] Nanyang Technological University, Singapore
alexandre.duc@epfl.ch
{ntu.guo,thomas.peyrin}@gmail.com
wl@pmail.ntu.edu.sg

Abstract. We analyze the internal permutations of KECCAK, one of the NIST SHA-3 competition finalists, in regard to differential properties. By carefully studying the elements composing those permutations, we are able to derive most of the best known differential paths for up to 5 rounds. We use these differential paths in a rebound attack setting and adapt this powerful freedom degrees utilization in order to derive distinguishers for up to 8 rounds of the internal permutations of the submitted version of KECCAK. The complexity of the 8 round distinguisher is $2^{491.47}$. Our results have been implemented and verified experimentally on a small version of KECCAK. This is currently the best known differential attack against the internal permutations of KECCAK.

New attacks on Keccak-224 and Keccak-256

Itai Dinur[1], Orr Dunkelman[1,2] and Adi Shamir[1]

[1] Computer Science department, The Weizmann Institute, Rehovot, Israel
[2] Computer Science Department, University of Haifa, Israel

Abstract. The Keccak hash function is one of the five finalists in NIST's SHA-3 competition, and so far it showed remarkable resistance against practical collision finding attacks: After several years of cryptanalysis and a lot of effort, the largest number of Keccak rounds for which actual collisions were found was only 2. In this paper we develop improved collision finding techniques which enable us to double this number. More precisely, we can now find within a few minutes on a single PC actual collisions in standard Keccak-224 and Keccak-256, where the only modification is to reduce their number of rounds to 4. When we apply our techniques to 5-round Keccak, we can get in a few days excellent near collisions, where the Hamming distance is 5 in the case of Keccak-224 and 10 in the case of Keccak-256. Our new attack combines differential and algebraic techniques, and uses the fact that each round of Keccak is only a quadratic mapping in order to efficiently find pairs of messages which follow a high probability differential characteristic.

Improved Zero-sum Distinguisher for Full Round KECCAK-f Permutation

Ming Duan[12] and Xuejia Lai[1]

[1] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, China.*

[2] *Basic Courses Department, University of Foreign Language, Luoyang, China.*

Abstract

KECCAK is one of the five hash functions selected for the final round of the SHA-3 competition and its inner primitive is a permutation called KECCAK-f. In this paper, we find that for the inverse of the only one nonlinear transformation of KECCAK-f, the algebraic degrees of any output coordinate and of the product of any two output coordinates are both 3 and also 2 less than its size 5. Combining the observation with a proposition from an upper bound on the degree of iterated permutations, we improve the zero-sum distinguisher of full 24 rounds KECCAK-f permutation by lowering the size of the zero-sum partition from 2^{1590} to 2^{1579}.

H(K | | M)

is a secure MAC with SHA-3

not with SHA-2 (length extension)

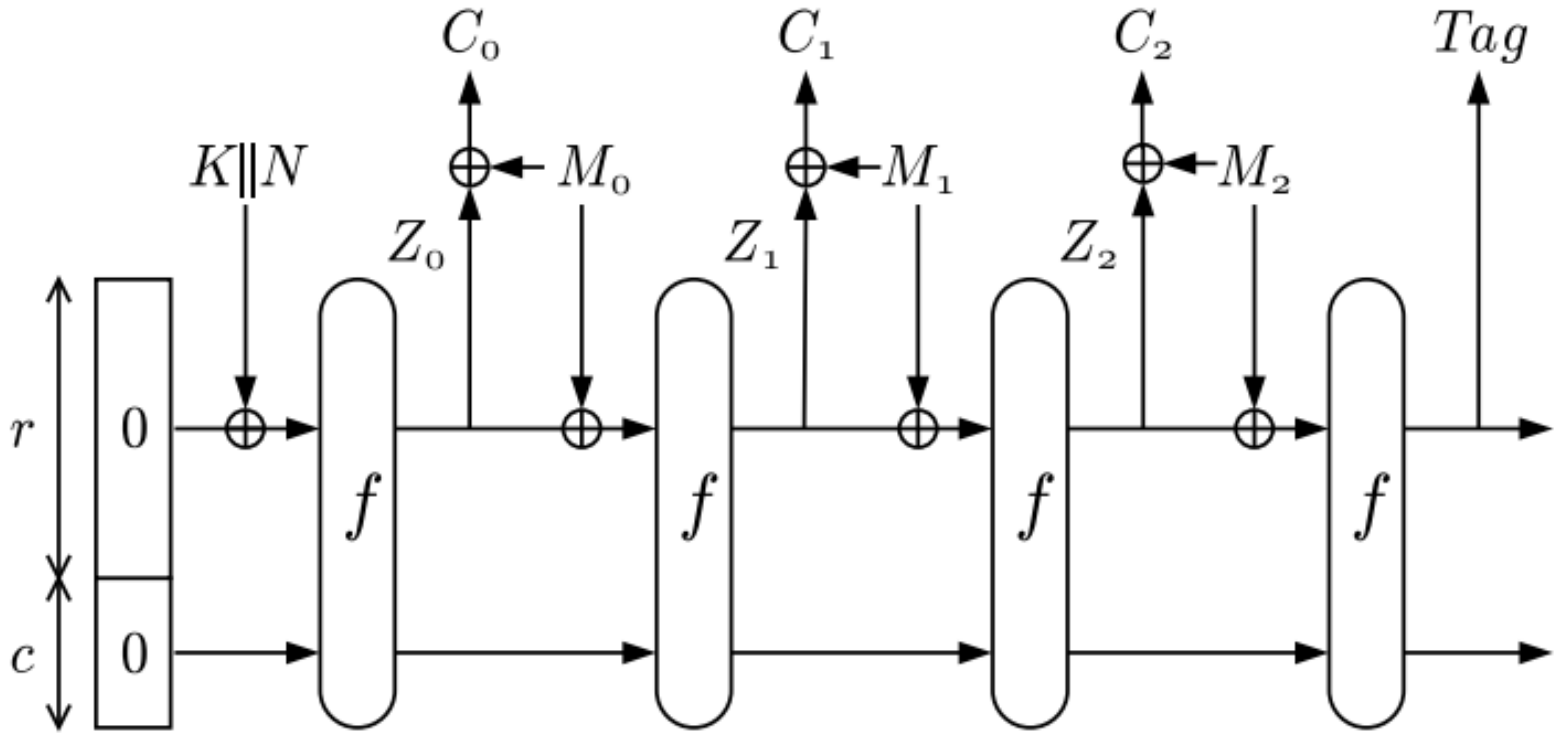
But key can be recovered if state leaks…
(unlike with BLAKE or Skein)

SHA-3: should we care?

“The only ordering implied in sha2 vs sha3 is when they were designed; we explicitly are ***not* telling people they should move from sha2 to sha3**. The two standards will coexist.”

John Kelsey (NIST)

Keccak's sponge construction can be used for RNG, MAC, authenticated encryption (***not*** *standardized by NIST)*



http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/Aug2010/documents/presentations/DAEMEN_SpongeDuplexSantaBarbaraSlides.pdf

What about BLAKE?

“BLAKE could be the Xenia T. of crypto!”

It’s not always the winner that wins…

BLAKE is the **fastest finalist** in software

In many applications hashing is not the speed bottleneck, but sometimes faster hashing means improved performance:

- Cloud storage integrity check
- File systems deduplication (e.g. ZFS)
- HIDS monitoring
- P2P integrity check

There's a reason why MD5, Tiger, or CRCs are still used…

BLAKE2

coming soon…

Thank you!

Matthew Green
@matthew_d_green

Please help me spread the meme that Keccak is pronounced 'ketchup'.

Reply Delete Favorite

33 RETWEETS | **4** FAVORITES