

The MD6 Hash Function

Ronald L. Rivest
MIT CSAIL



First SHA-3 Candidate Conference
Leuven, Belgium
February 26, 2009

MD6 Team

- ◆ Dan Bailey
- ◆ Sarah Cheng
- ◆ Christopher Crutchfield
- ◆ Yevgeniy Dodis
- ◆ Elliott Fleming
- ◆ Asif Khan
- ◆ Jayant Krishnamurthy
- ◆ Yuncheng Lin
- ◆ Leo Reyzin
- ◆ Emily Shen
- ◆ Jim Sukha
- ◆ Eran Tromer
- ◆ Yiqun Lisa Yin
- ◆ Juniper Networks
- ◆ Cilk Arts
- ◆ NSF

Outline

- ◆ Design considerations
- ◆ Mode of Operation
- ◆ Compression Function
- ◆ Implementations
- ◆ Security

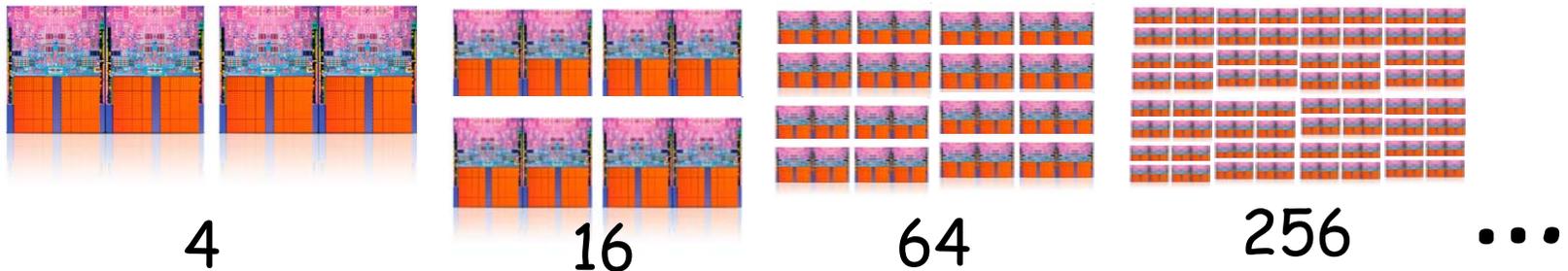
In response to recent attacks:

- ◆ (Differential attacks of Wang et al.)
- ◆ MD6 is *provably* resistant to standard differential attacks
- ◆ (SHA-3 should be, too!)

Design Considerations / Responses

Parallelism has arrived

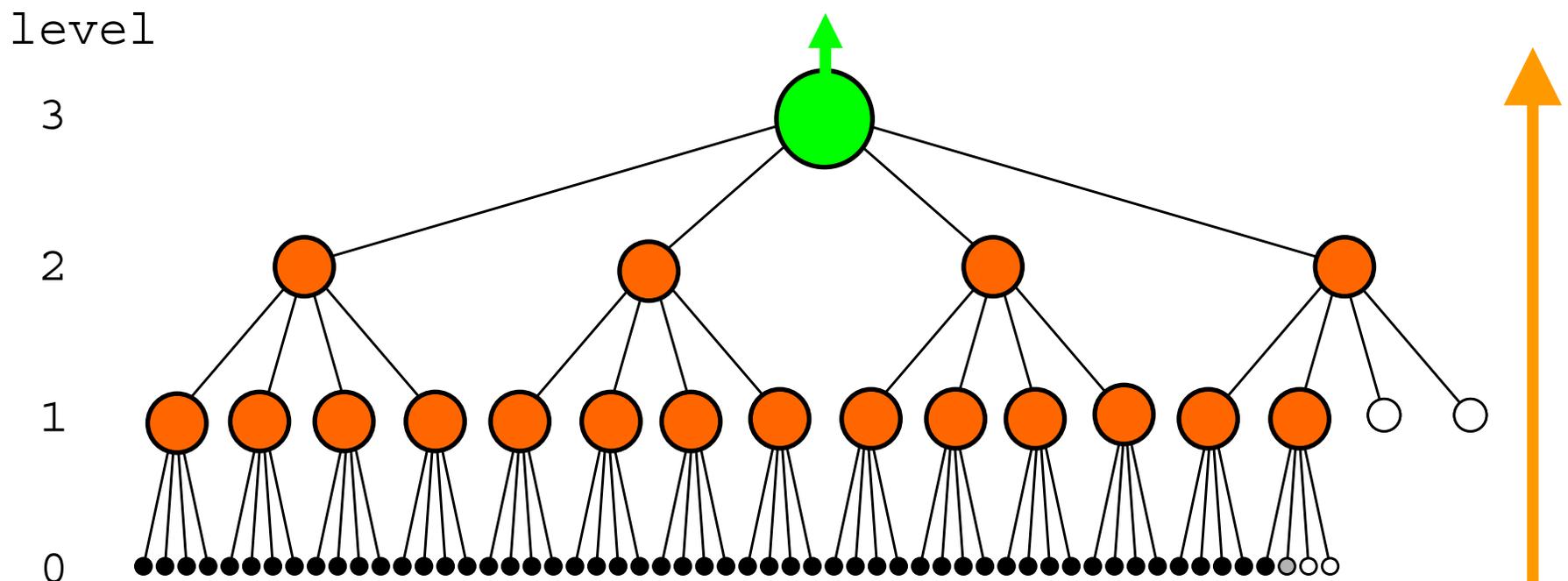
- ◆ Uniprocessors have “hit the wall”
 - Clock rates have *plateaued*
- ◆ Hundreds of cores coming soon to a desktop near you!



- ◆ SHA-3 should be *parallelizable!*

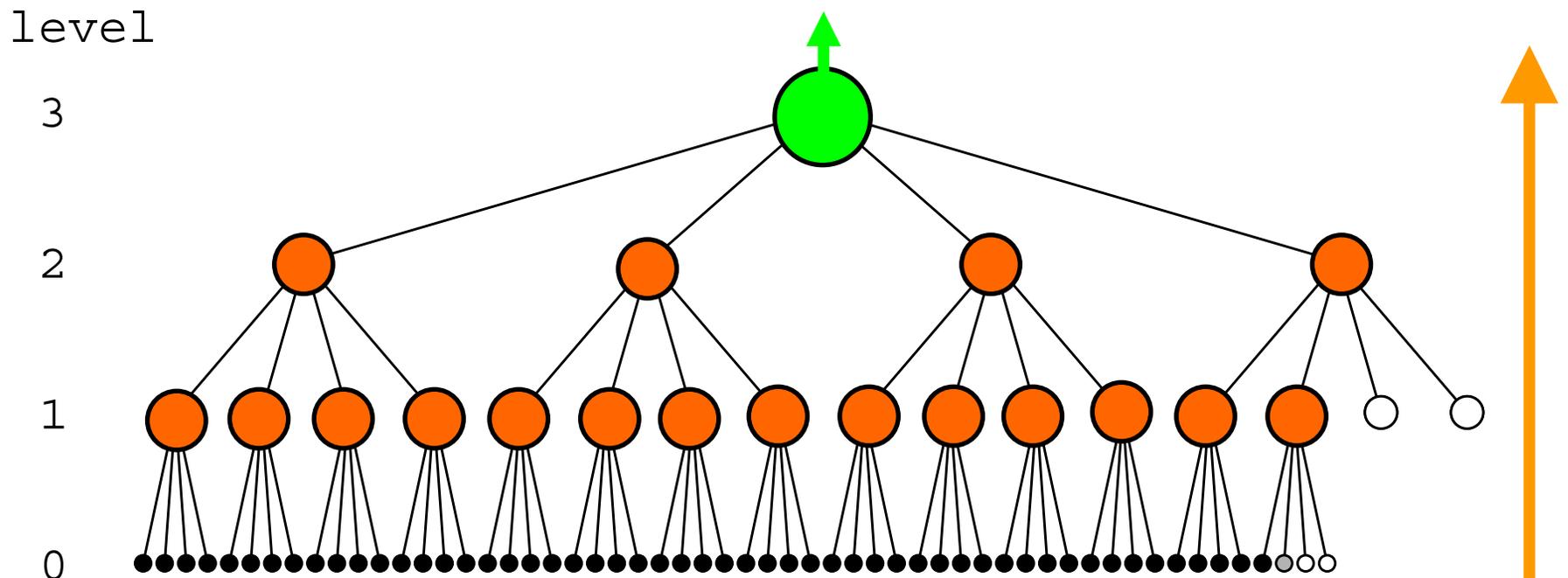
So... MD6 is tree-based

- ◆ Bottom-up tree-based mode of operation (like Merkle-tree)
- ◆ 4-to-1 compression ratio at each node



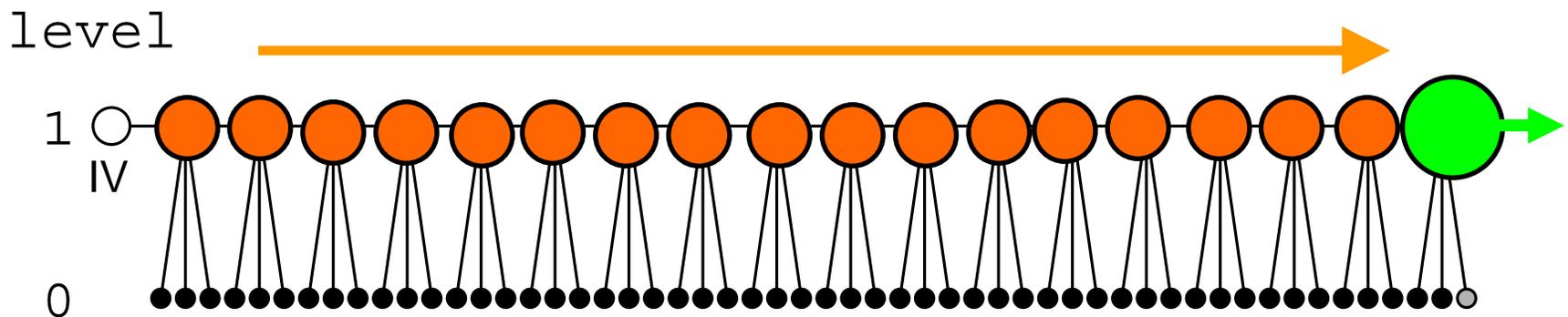
Which works very well in parallel

- ◆ Height is $\log_4(\text{number of nodes})$



For very tiny CPU's MD6 has...

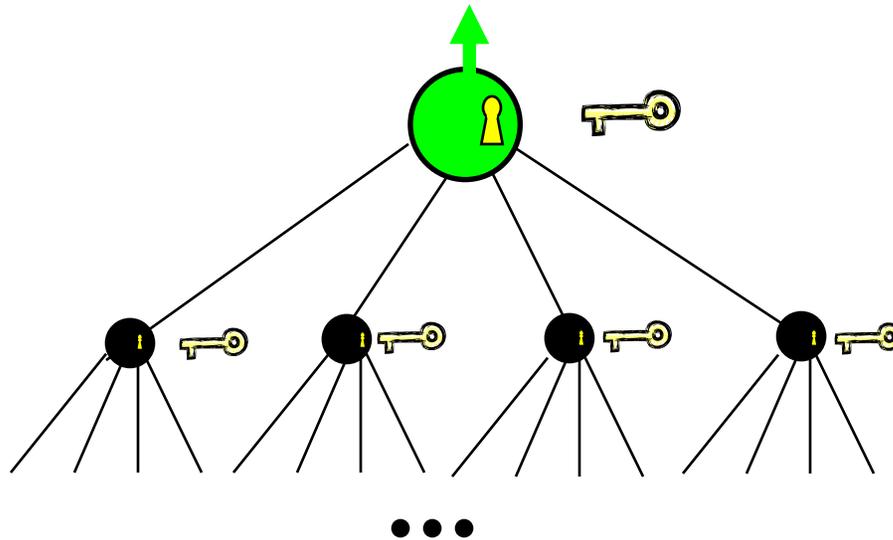
- ◆ Alternative sequential mode



- ◆ (Fits in 1KB RAM)

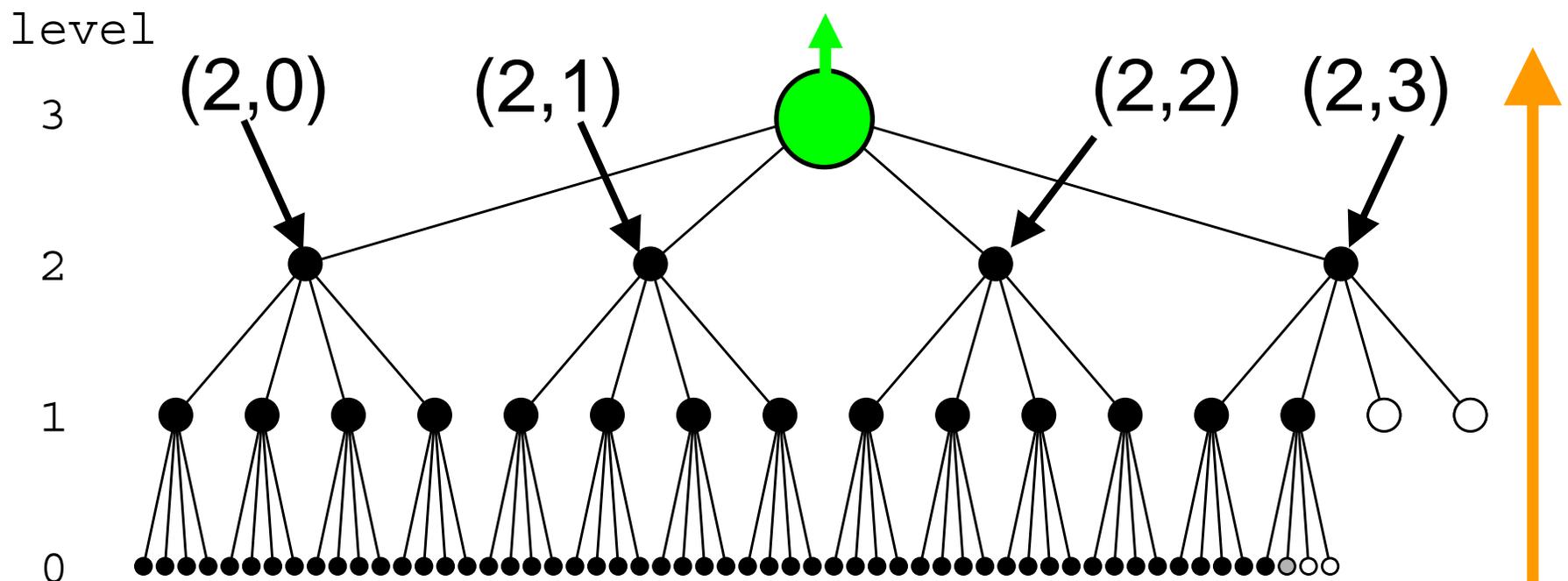
MD6 is *keyed*

- ◆ (For salt, MAC key, etc.)
- ◆ Key input K  of up to 512 bits
- ◆ K input to every compression function



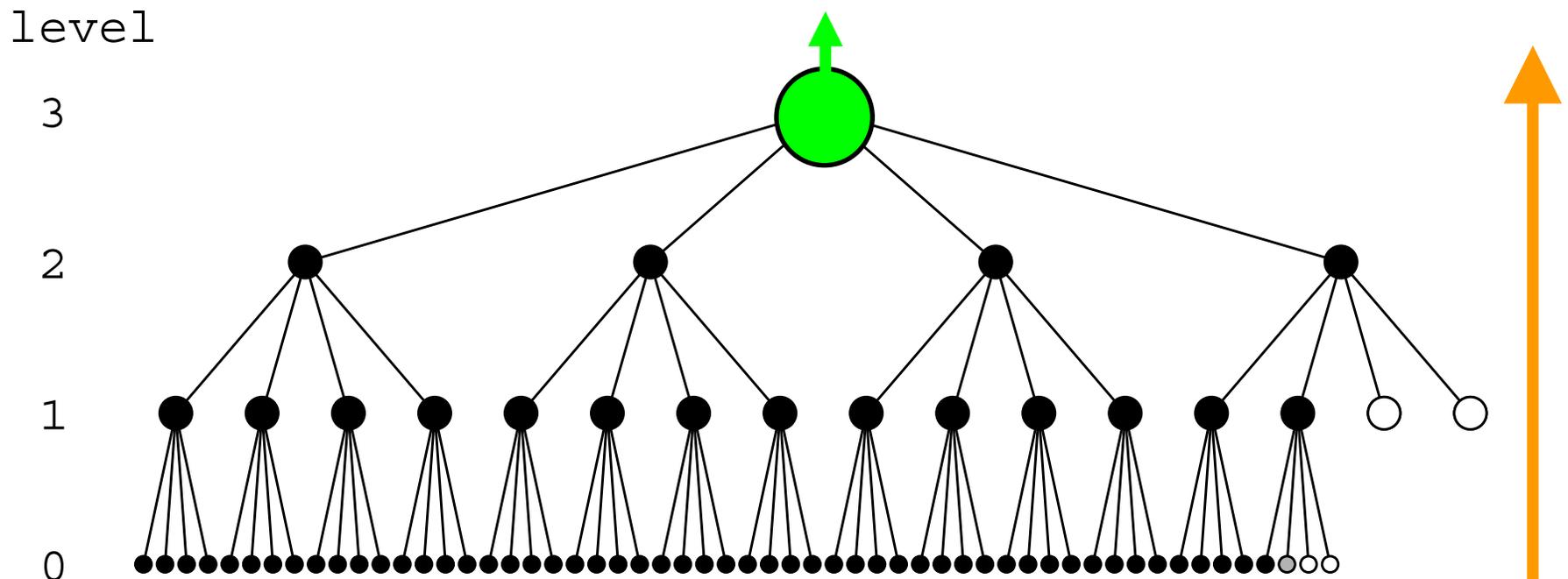
For “good hygiene” MD6 has:

- ◆ 1024-bit intermediate (chaining) values; root output chopped to desired length
- ◆ Location (level,index) input to each node



And the root is special!

- ◆ Compression function inputs “root bit” (z-bit or “green bit”) which is True only at root:



MD6 Compression function

To prevent side-channel attacks:

- ◆ MD6 uses only the following *safe* operations, on 64-bit words:

- XOR



- AND



- SHIFT by fixed amounts:

$x \gg r$



$x \ll \ell$



- ◆ (All SHA-3 candidates should be required to submit timings for a *safe* implementation! No table lookups!)

MD6 has variable number r of rounds

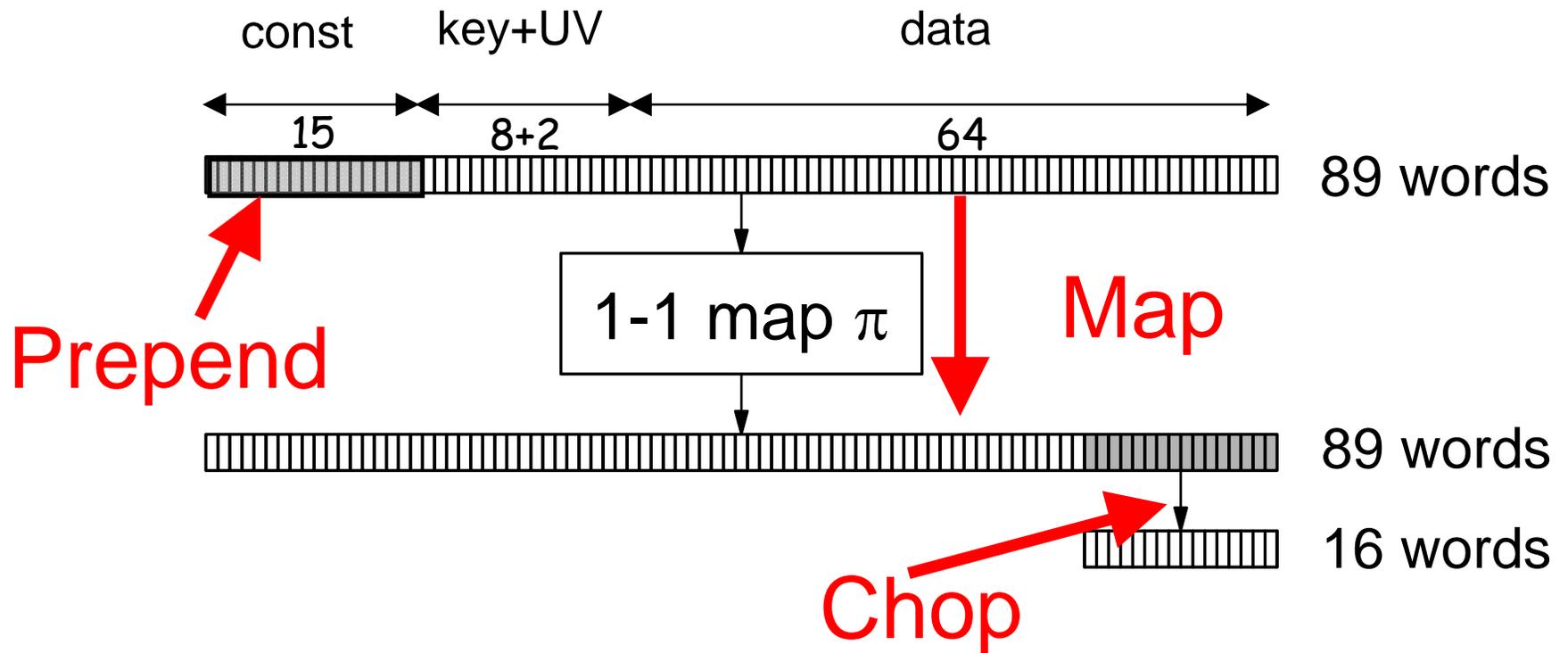
- ◆ A round is 16 steps.
- ◆ For output digest size of d bits, default is
 $r = 40 + (d/4)$

Digest size d	160	224	256	384	512
Rounds r	80	96	104	136	168

Compression function input

- ◆ 64 word (512 byte) data block
 - message, or up to 4 child chaining values
- ◆ 8 word (512 bit) key K
- ◆ 1 word location $U = (\text{level}, \text{index})$
- ◆ 1 word metadata V :
 - Padding amount, key length, z-bit, max tree height, digest output size d , number r of rounds.
- ◆ 74 words total

Prepend Constant + Map + Chop



Simple compression function:

Input: $A[0 .. 88]$ of $A[0 .. 16r + 88]$

for $i = 89$ **to** $16r + 88$:

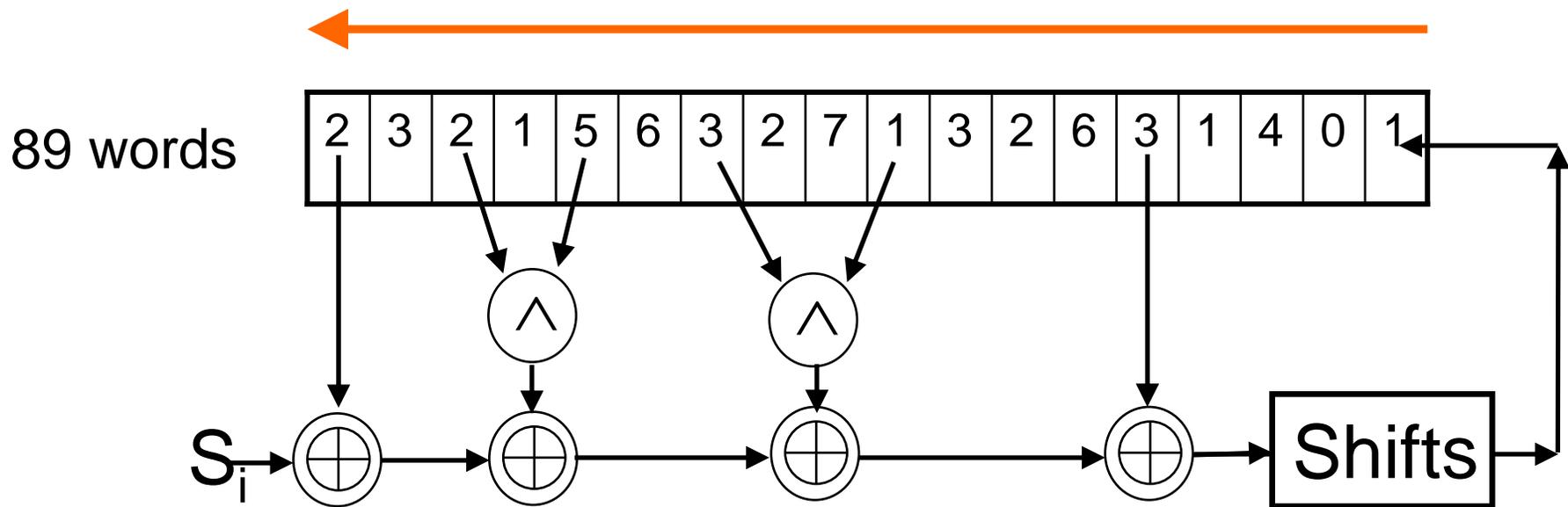
$$\begin{aligned}x &= S_i \oplus A[i-17] \oplus A[i-89] \\ &\quad \oplus (A[i-18] \wedge A[i-21]) \\ &\quad \oplus (A[i-31] \wedge A[i-67])\end{aligned}$$

$$x = x \oplus (x \gg r_i)$$

$$A[i] = x \oplus (x \ll l_i)$$

return $A[16r + 73 .. 16r + 88]$

712 byte shift-reg implementation

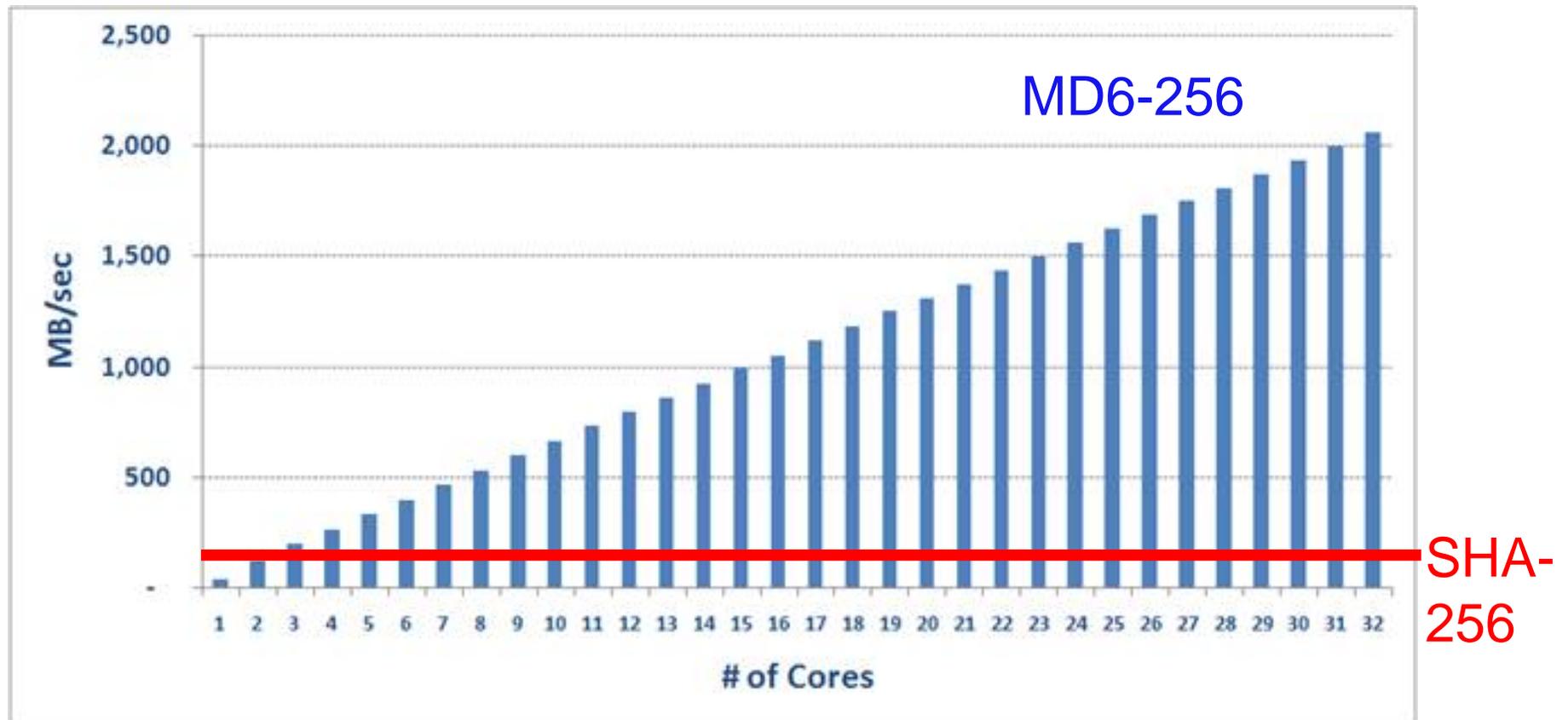


Implementations

NIST SHA-3 Reference Platforms

	32-bit	64-bit
MD6-160	54 cpb	24 cpb
MD6-224	63 cpb	29 cpb
MD6-256	68 cpb	31 cpb
MD6-384	87 cpb	40 cpb
MD6-512	106 cpb	48 cpb
SHA-512	63 cpb	13 cpb

Multicore efficiency > 2GB/sec !

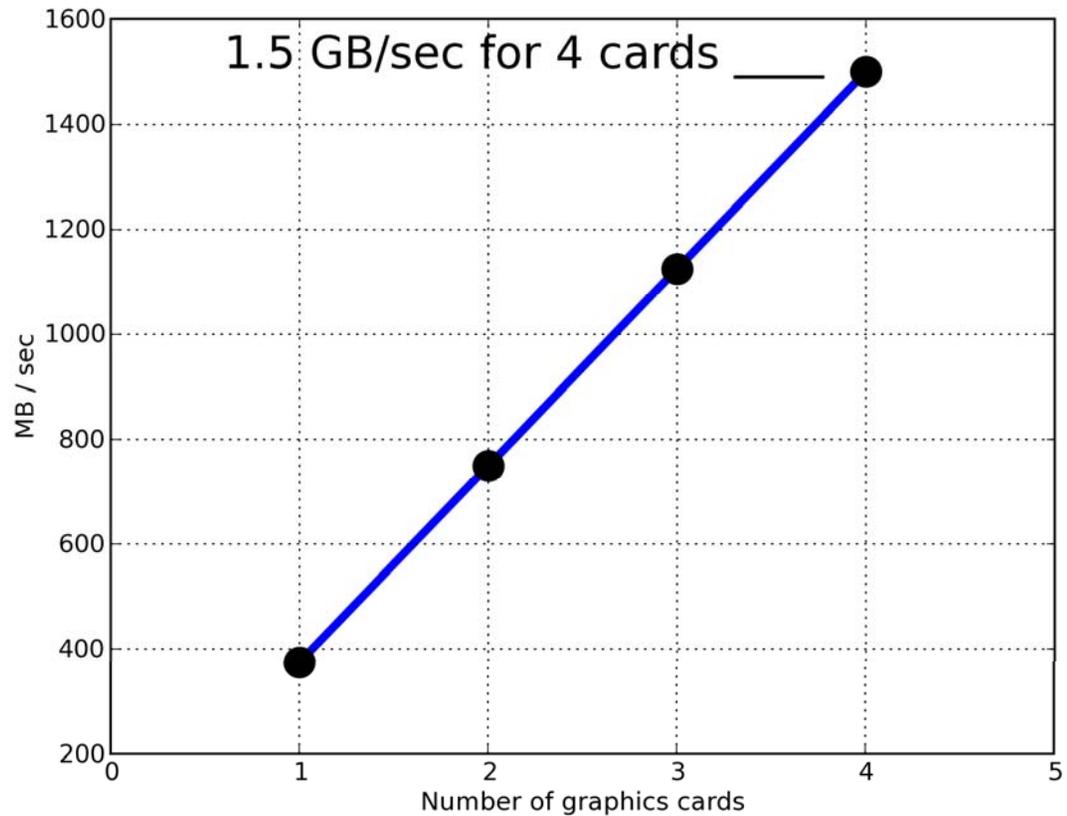


This is real data, courtesy of *Cilk Arts*!

Efficiency on a GPU



- ◆ Standard
\$100
NVidia
GPU
- ◆ 375
MB/sec
on one
card



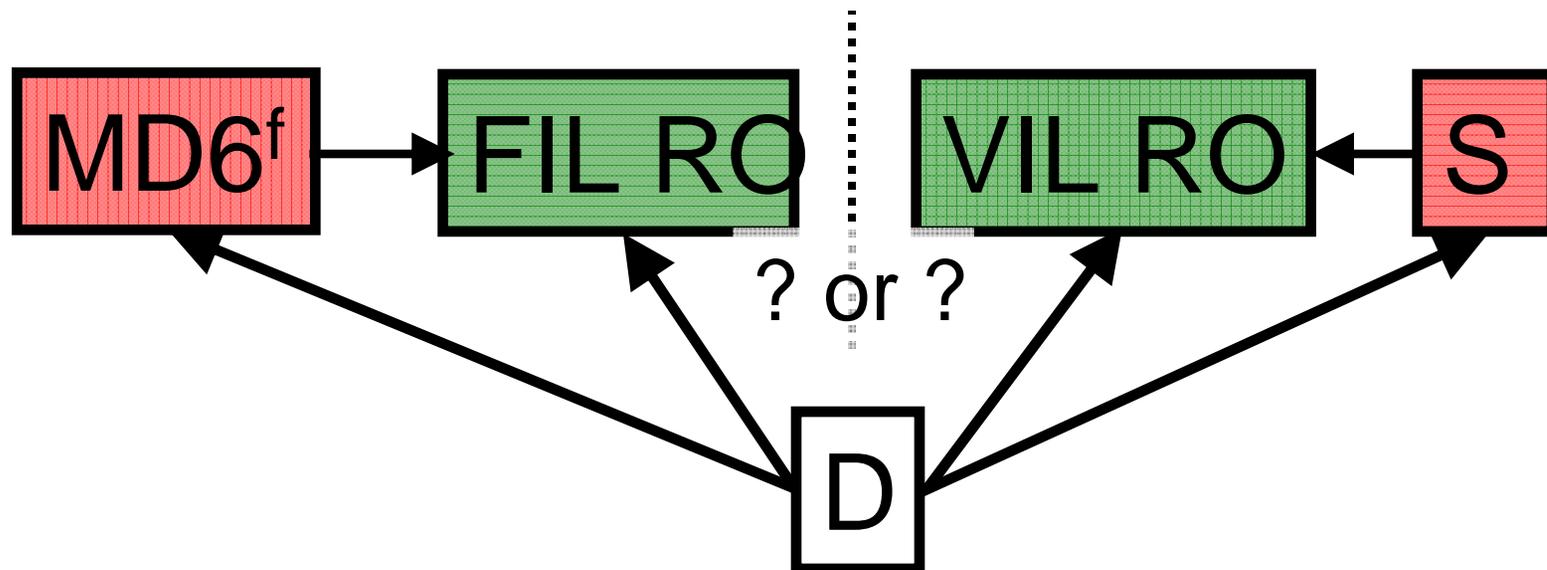
Security

Property-Preservations

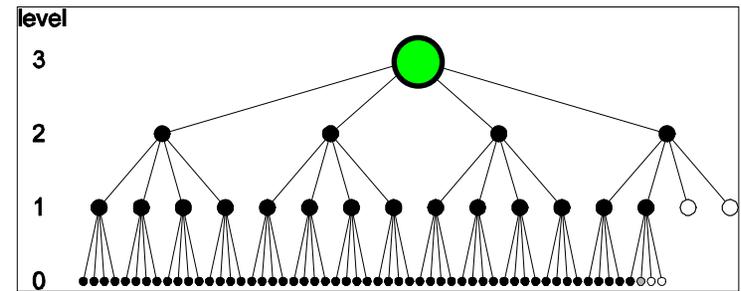
- ◆ **Theorem.** If f is collision-resistant, then MD6^f is collision-resistant.
- ◆ **Theorem.** If f is preimage-resistant, then MD6^f is preimage-resistant.
- ◆ **Theorem.** If f is a FIL-PRF, then MD6^f is a VIL-PRF.
- ◆ **Theorem.** If f is a FIL-MAC and root node effectively uses distinct random key (due to z -bit), then MD6^f is a VIL-MAC.
- ◆ (See thesis by Crutchfield.)

Indifferentiability (Maurer et al. '04)

- ◆ Variant notion of indistinguishability appropriate when distinguisher has access to inner component (e.g. mode of operation $MD6^f$ / comp. fn f).

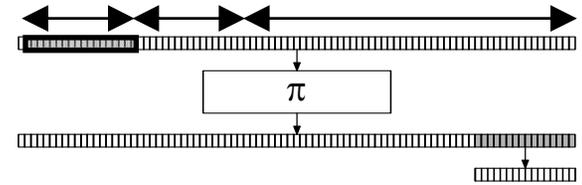


Indifferentiability (I)



- ◆ **Theorem.** The MD6 mode of operation is indifferentiable from a random oracle.
- ◆ **Proof:** Construct simulator for compression function that makes it consistent with any VIL RO and MD6 mode of operation...
- ◆ (All SHA-3 candidates should have such a result known for them!)

Indifferentiability (II)



- ◆ **Theorem.** MD6 compression function f^π is indifferentiable from a FIL random oracle (with respect to random permutation π).
- ◆ **Proof:** Construct simulator S for π and π^{-1} that makes it consistent with FIL RO and comp. fn. construction.

Differential attacks don't work

- ◆ **Theorem.** *Any standard differential attack has less chance of finding collision than standard birthday attack.*

Differential attacks (cont.)

- ◆ Compare birthday bound BB with our lower bound LB on work for any standard differential attack.
- ◆ (Gives adversary fifteen rounds for message modification, etc.)
- ◆ These bounds can be improved...

d	r	BB	LB
160	80	2^{80}	2^{104}
224	96	2^{112}	2^{130}
256	104	2^{128}	2^{150}
384	136	2^{192}	2^{208}
512	168	2^{256}	2^{260}

Attacks

- ◆ Collision known for 16 rounds [Khazaei]
- ◆ Distinguishable from RO for 18 rounds [Aumasson et al.]
- ◆ Key recovery for 14 rounds [Aumasson et al.]
- ◆ Fixing $Q=0$, can distinguish up to 33 rounds [Khovratovich]
- ◆ Fixing $S=0$, can distinguish up to 66 rounds [Aumasson et al.]

Choosing number of rounds

- ◆ For digest sizes 224 ... 512 , MD6 has 80 ... 168 rounds; these defaults are conservative (intentionally); MD6 may well be secure at 40 rounds (which gives 12 cpb for 64-bit platform).
- ◆ Default allows *proof* of resistance to differential cryptanalysis; these proofs may get better!

Summary

- ◆ MD6 is:
 - Arguably secure against known attacks (including differential attacks)
 - Relatively simple
 - Highly parallelizable
 - Reasonably efficient

THE END

MD6

03744327e1e959fbdcdf7331e959cb2c28101166