The MD6 Hash Function (aka "Pumpkin Hash")

Ronald L. Rivest

MIT CSAIL

CRYPTO 2008

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

- Introduction
- Design considerations
- Mode of Operation
- Compression Function
- Software Implementations
- Hardware Implementations
- Security Analysis

MD5 was designed in 1991…

- Same year WWW announced…
- Clock rates were 33MHz…
- Requirements:
  - $\{0,1\}^* \longrightarrow \{0,1\}^d$ for digest size d
  - Collision-resistance
  - Preimage resistance
  - Pseudorandomness
- What’s happened since then?
- Lots…  
- What should a hash function --- MD6 --- look like today?

NIST SHA-3 competition!

- Input: 0 to 2^{64}-1 bits, size not known in advance
- Output sizes 224, 256, 384, 512 bits
- Collision-resistance, preimage resistance, second preimage resistance, pseudorandomness, …
- Simplicity, flexibility, efficiency, …
- Due Halloween ‘08

Design Considerations / Responses

Wang et al. break MD5 (2004)

- Differential cryptanalysis (re)discovered by Biham and Shamir (1990). Considers step-by-step ``difference'' (XOR) between two computations…
- Applied first to block ciphers (DES)…
- Used by Wang et al. to break collision-resistance of MD5
- Many other hash functions broken similarly; others may be vulnerable…

So… MD6 is…

- *provably* resistant to differential attacks (more on this later…)


Memory is now ``plentiful”… 

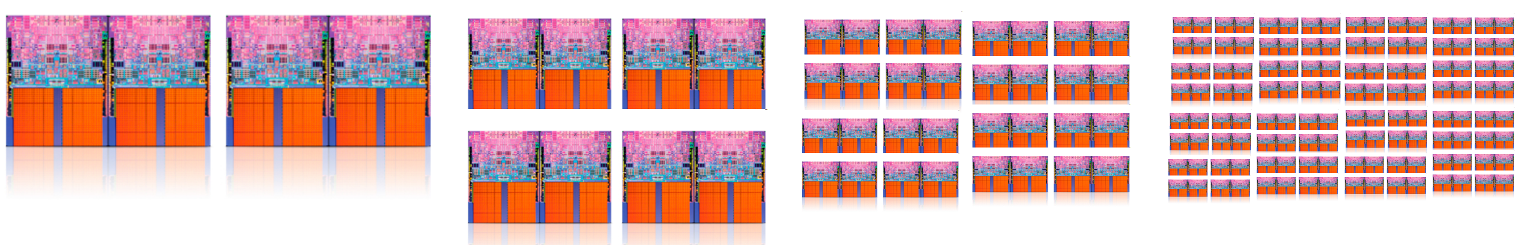
- Memory capacities have increased 60% *per year* since 1991
- Chips have 1000 times as much memory as they did in 1991
- Even ``embedded processors” typically have at least 1KB of RAM

So… MD6 has…

- *Large* input message block size: 512 *bytes* (not 512 bits)
- This has many advantages…

Parallelism has arrived

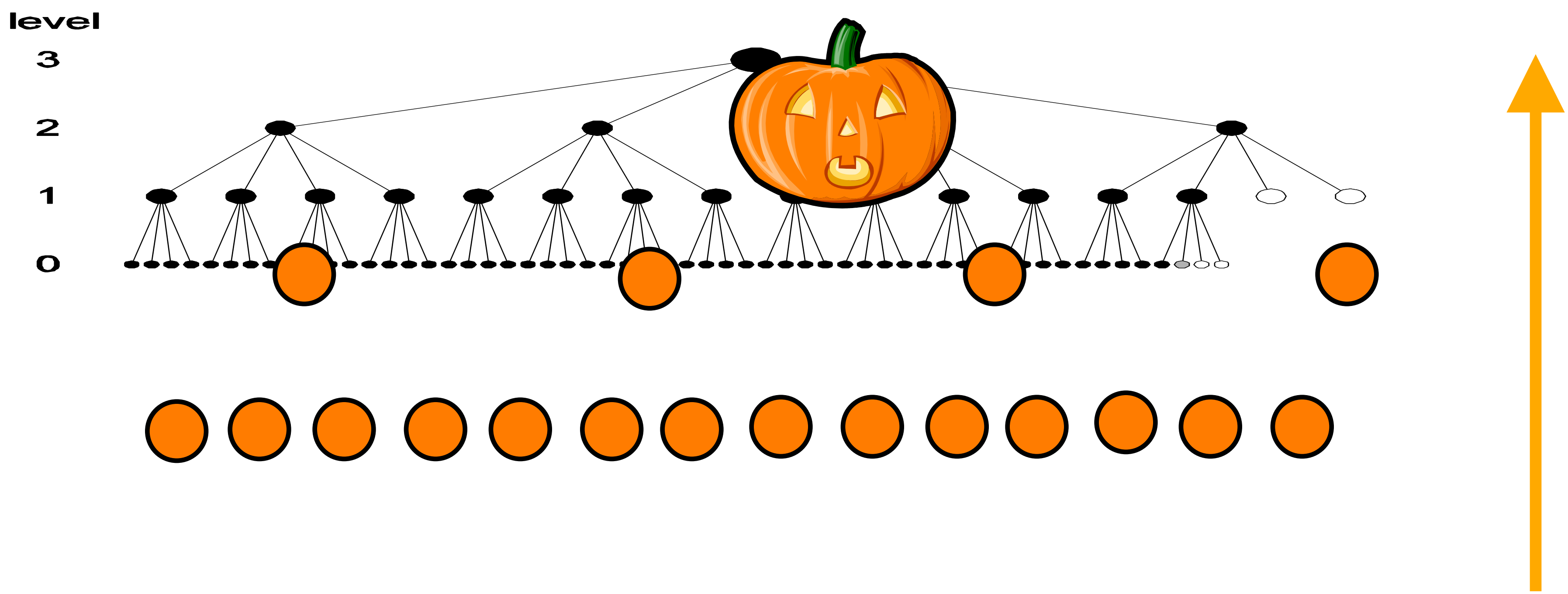
- Uniprocessors have "hit the wall"
  - Clock rates have *plateaued*, since power usage is quadratic or cubic with clock rate:
    $P = VI = V^2/R = O(\ freq^2\ )$ (roughly)
- Instead, *number of cores* will double with each generation: tens, hundreds (thousands!) of cores coming soon



4   16   64   256   …

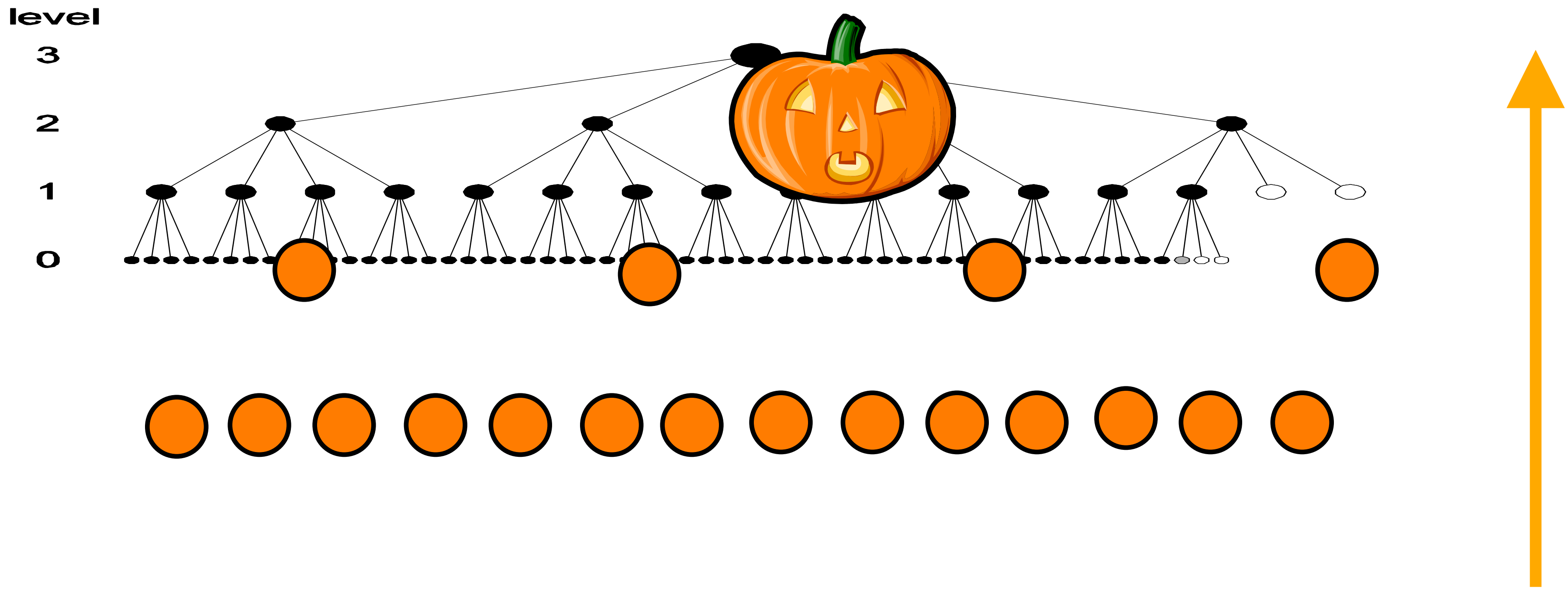
So… MD6 has…

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



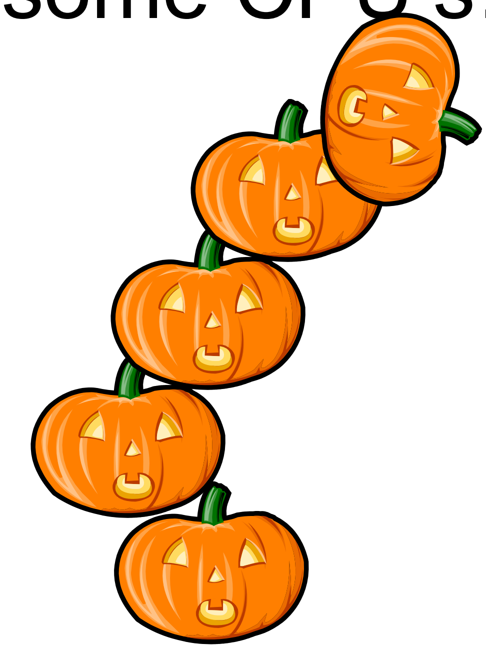
Which works very well in parallel

- Height is $\log_4$(number of nodes)



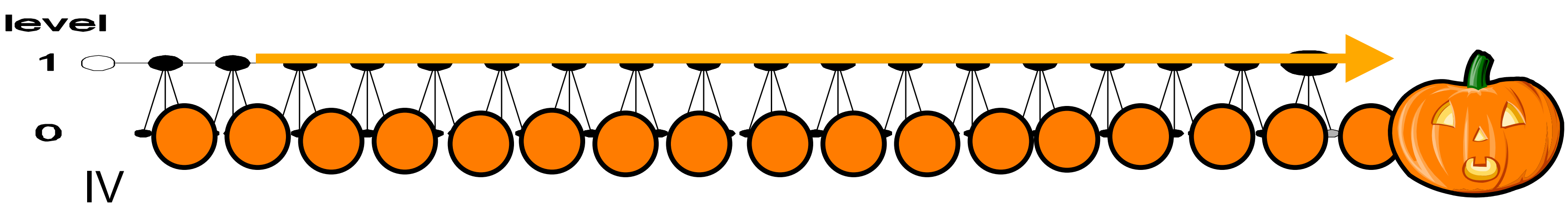
But… most CPU’s are small… 

- Most biomass is bacteria…
- Storage proportional to tree height may be too much for some CPU’s…



So… MD6 has…

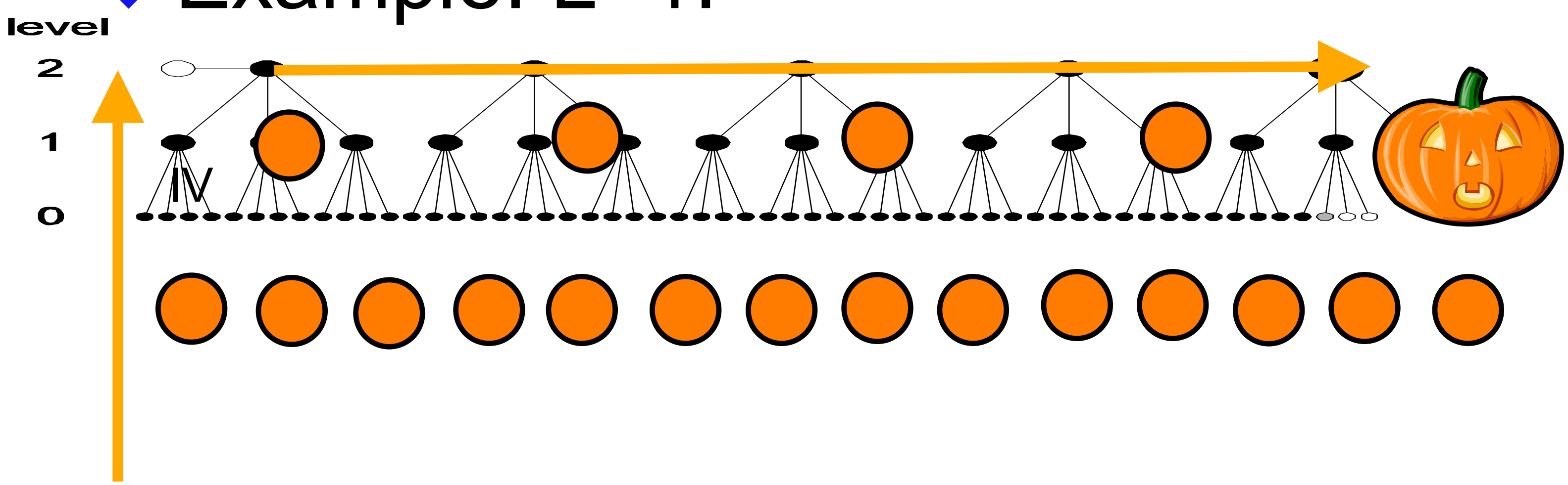
- Alternative sequential mode



- (Fits in 1KB RAM)

Actually, MD6 has…

- a smooth sequence of alternative modes: from purely sequential to purely hierarchical… L parallel layers followed by a sequential layer, $0 \leq L \leq 64$
- Example: L=1:

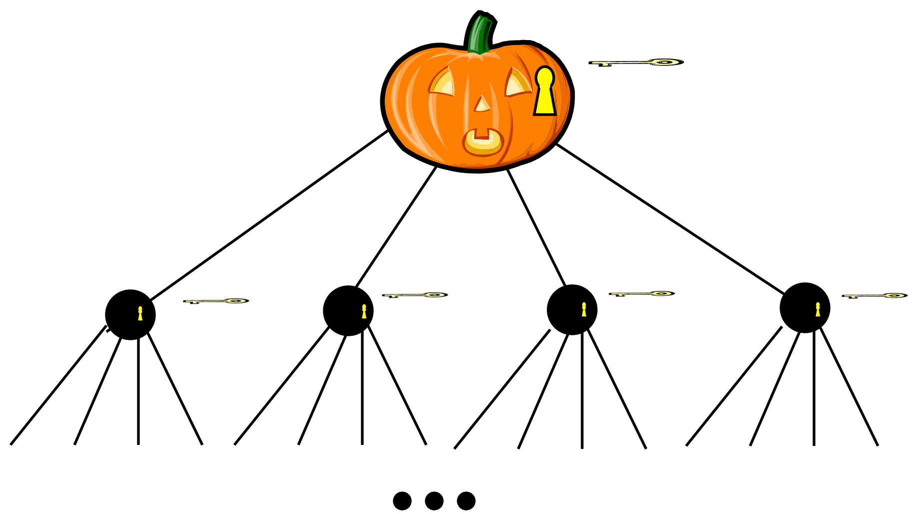


Hash functions often ``keyed'' 

- Salt for password, key for MAC, variability for key derivation, theoretical soundness, etc…
- Current modes are "post-hoc"

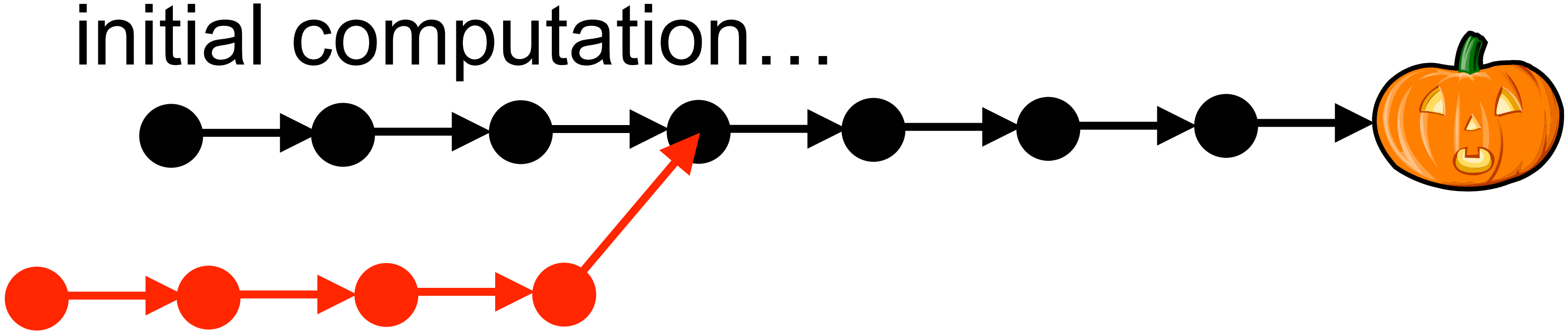
So… MD6 has..

- *Key input K* of up to 512 bits
- K is input to *every* compression function



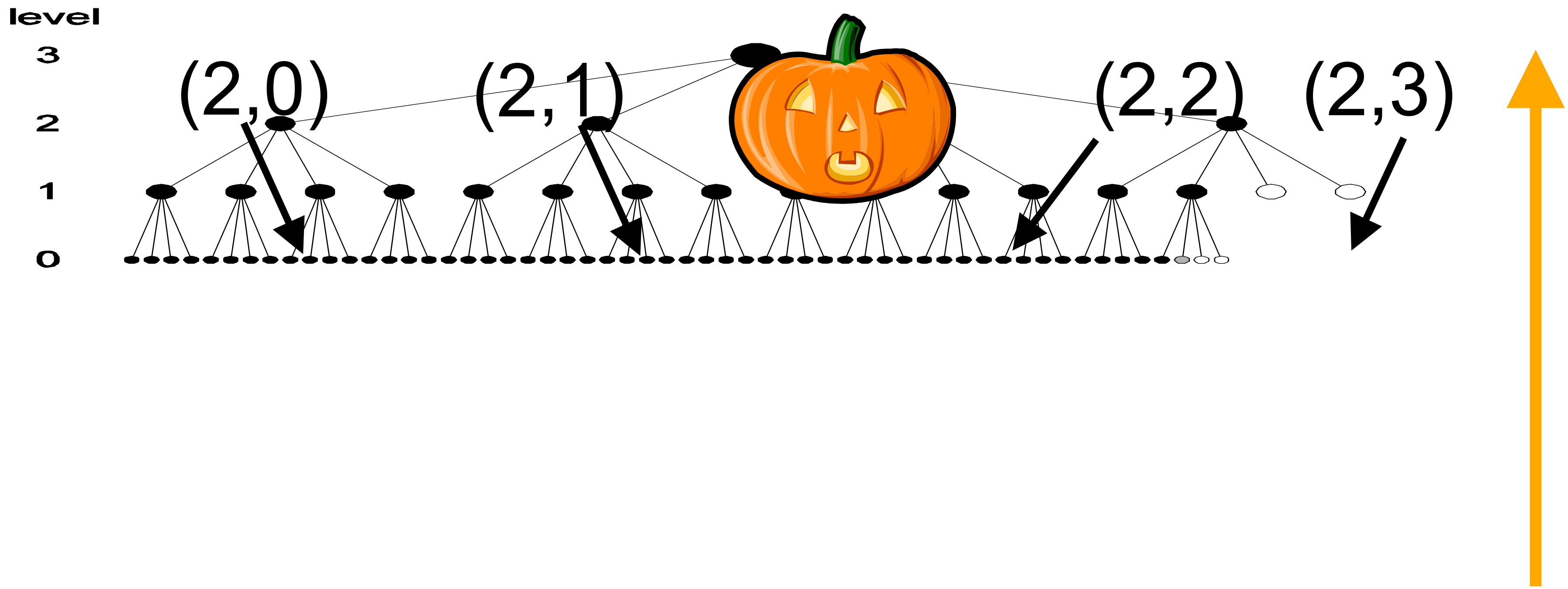
Generate-and-paste attacks 

- Kelsey and Schneier (2004), Joux (2004), …
- Generate sub-hash and fit it in *somewhere*
- Has advantage proportional to size of initial computation…



So… MD6 has…

- 1024-bit intermediate (chaining) values
- root truncated to desired final length
- Location (level,index) input to each node



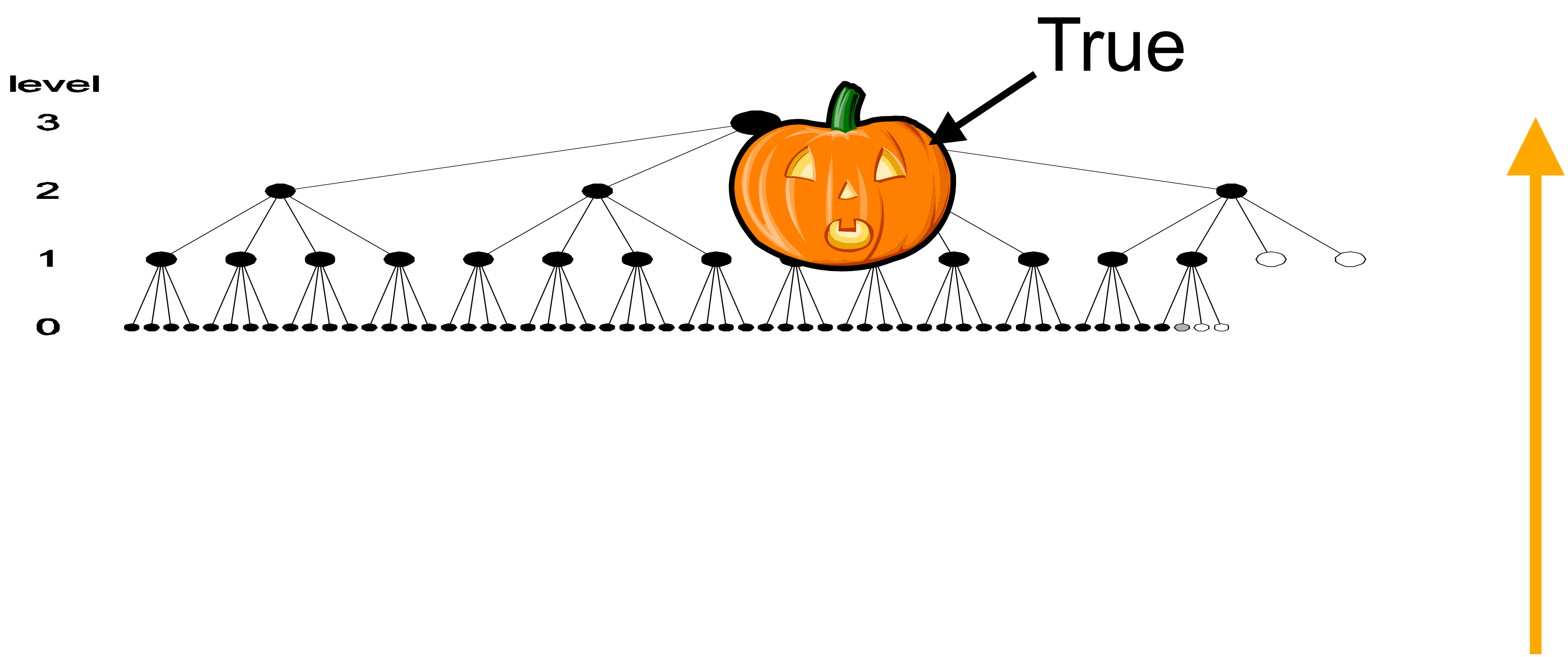
Extension attacks… 

- Hash of one message useful to compute hash of another message (especially if keyed):

$$H(\ K \,||\, A \,||\, B\ ) = H(\ H(\ K \,||\, A) \,||\, B\ )$$

So… MD6 has…

- ``Root bit’’ (aka “z-bit” or “pumpkin bit”) input to each compression function:



Side-channel attacks 

- Timing attacks, cache attacks…
- Operations with data-dependent timing or data-dependent resource usage can produce vulnerabilities.
- This includes data-dependent rotations, *table lookups* (S-boxes), some complex operations (e.g. multiplications), …

So… MD6 uses…

- Operations on 64-bit words
- The following operations *only:*
  - XOR $\oplus$
  - AND $\wedge$
  - SHIFT by fixed amounts:

    x >> r >>

    x << l <<

Security needs vary…

- Already recognized by having different digest lengths d   (for MD6: $1 \leq d \leq 512$)
- But it is useful to have reduced-strength versions for analysis, simple applications, or different points on speed/security curve.

So… MD6 has ..

- A variable number r of rounds.
  (Each round is 16 steps.)
- Default r depends on digest size d :

$$r = 40 + (d/4)$$

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

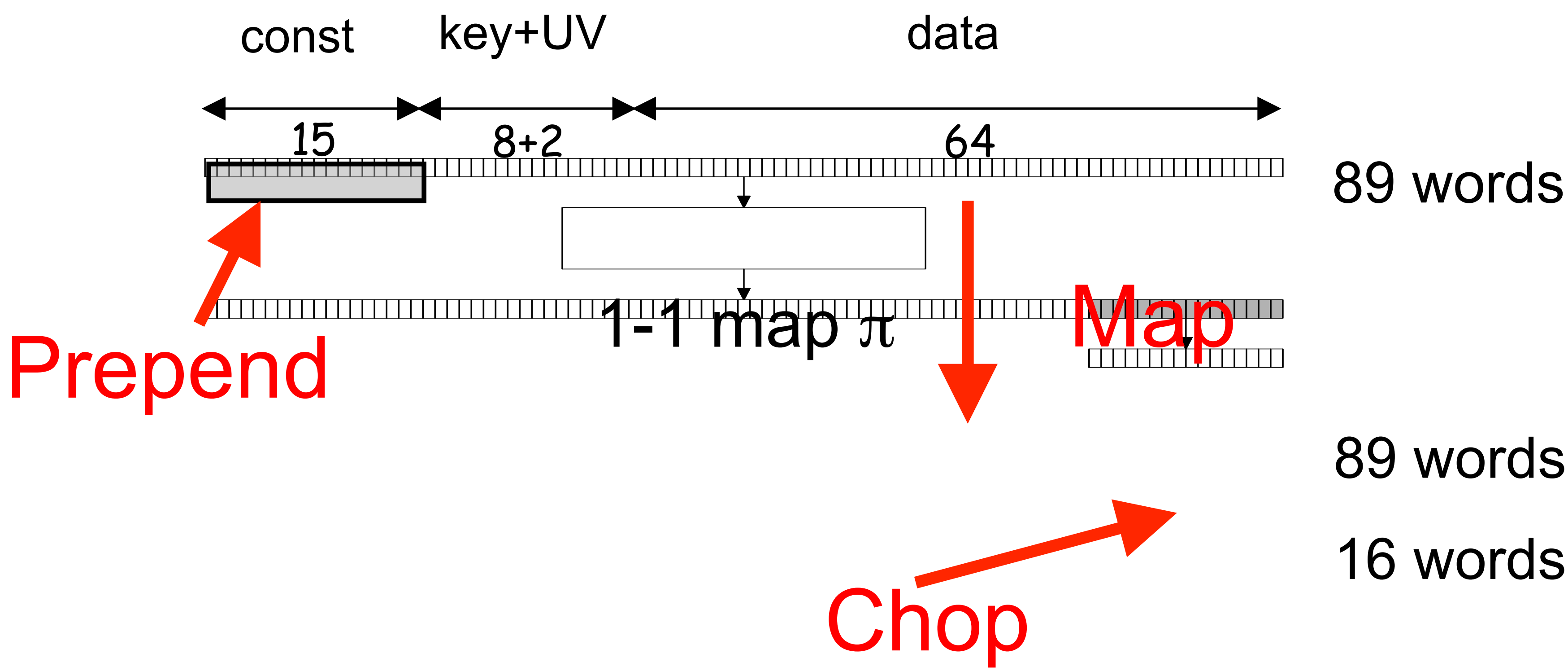
- But r is also an (optional) input.

MD6 Compression function

Compression function inputs

- 64 word (512 byte) data block
  - message, or chaining values
- 8 word (512 bit) key K
- 1 word U = (level, index)
- 1 word V = parameters:
  - Data padding amount
  - Key length ($0 \leq$ keylen $\leq$ 64 bytes)
  - z-bit (aka ``root bit’’ aka``pumpkin bit”)
  - L (mode of operation height-limit)
  - digest size d (in bits)
  - 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** 16 r + 88 :

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

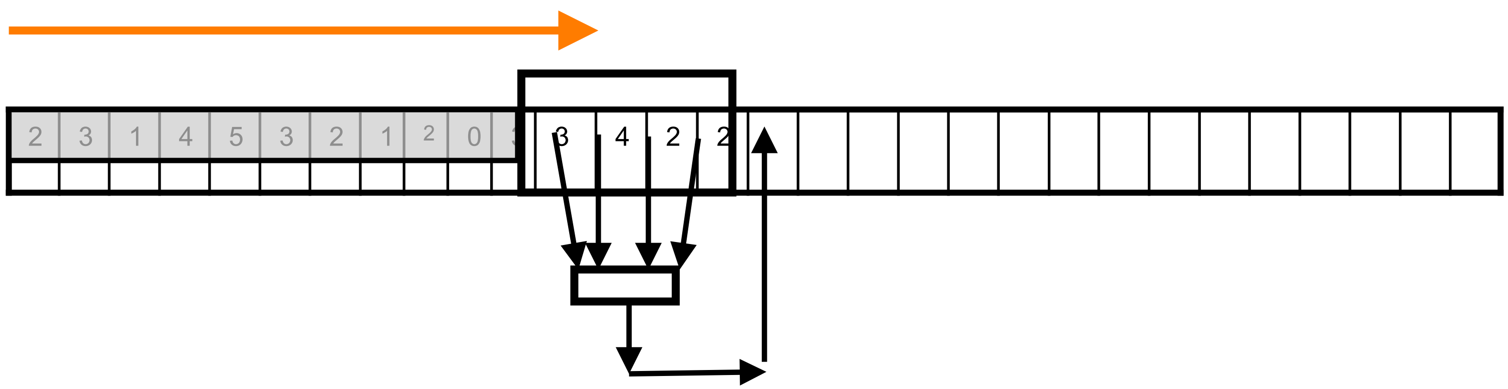
return A[16r + 73 .. 16r + 88]

Constants

- Taps 17, 18, 21, 31, 67 optimize diffusion
- Constants S_i defined by simple recurrence; change at end of each 16-step round
- Shift amounts repeat each round (best diffusion of 1,000,000 such tables):

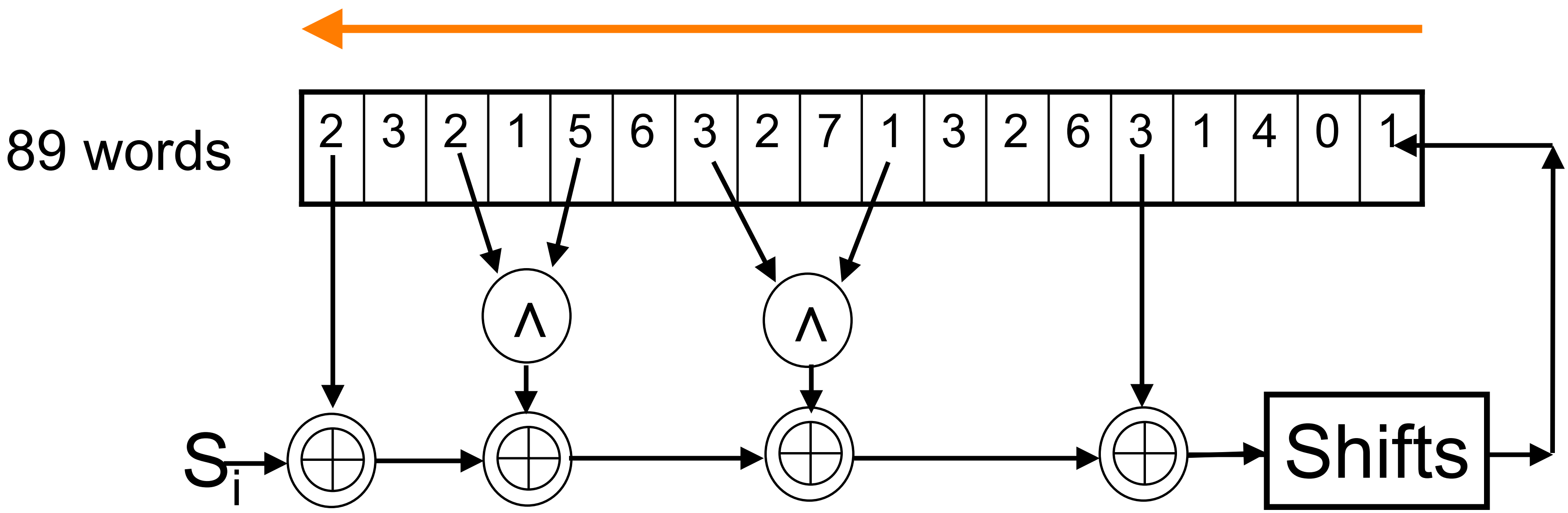
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r_i	10	5	13	10	11	12	2	7	14	15	7	13	11	7	6	12
l_i	11	24	9	16	15	9	27	15	6	2	29	8	15	5	31	9

Large Memory (sliding window)



- Array of 16r + 89 64-bit words.
- Each computed as function of preceding 89 words.
- Last 16 words computed are output.

Small memory (shift register)



- Shift-register of 89 words (712 bytes)
- Data moves right to left

Software Implementations

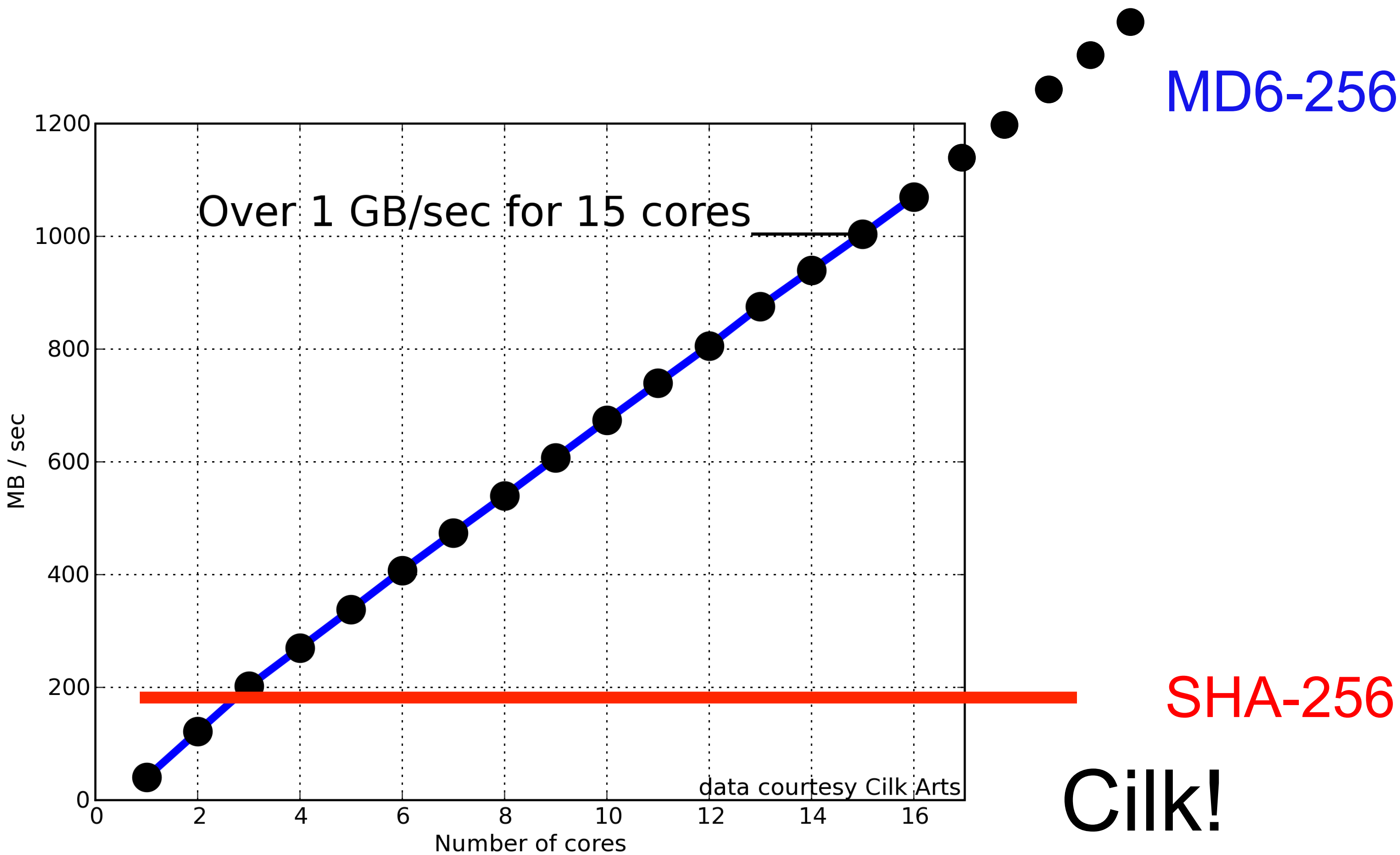
Software implementations

- Simplicity of MD6:
  - *Same* implementation for all digest sizes.
  - *Same* implementation for SHA-3 Reference or SHA-3 Optimized Versions.
  - Only optimization is *loop-unrolling* (16 steps within one round).

NIST SHA-3 Reference Platforms

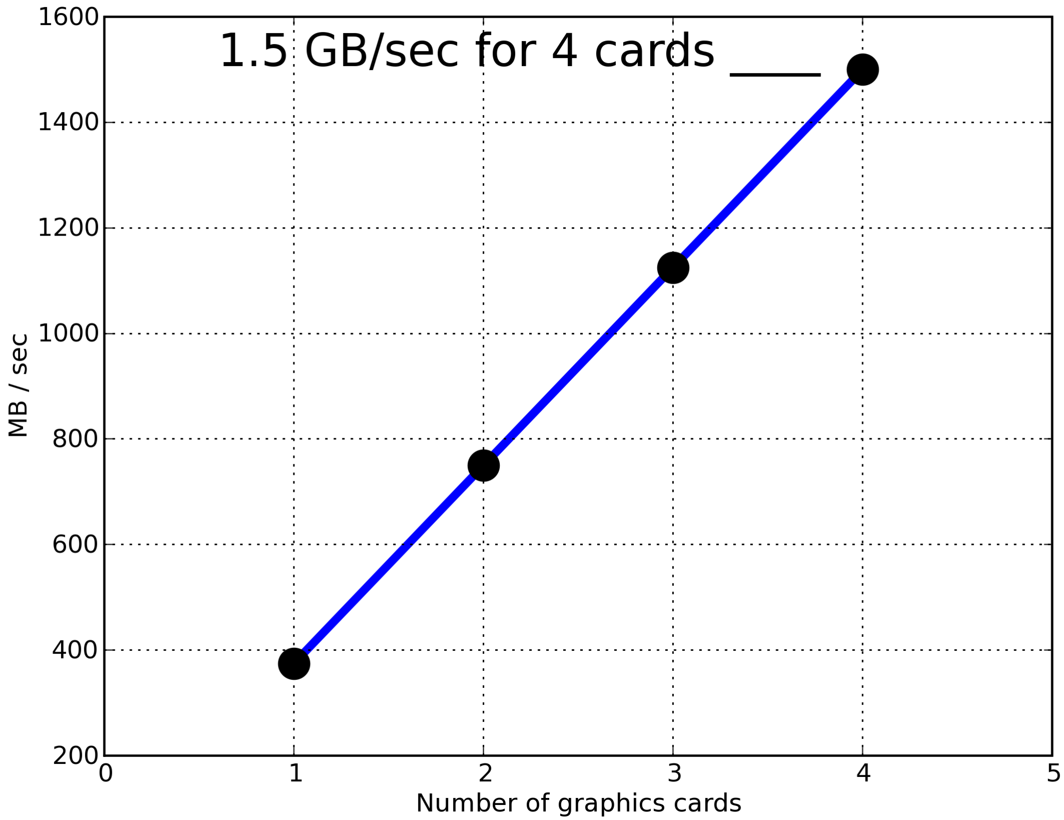
	32-bit	**64-bit**
MD6-160	44 MB/sec	97 MB/sec
MD6-224	38 MB/sec	82 MB/sec
MD6-256	35 MB/sec	77 MB/sec
MD6-384	27 MB/sec	59 MB/sec
MD6-512	22 MB/sec	49 MB/sec
SHA-512	38 MB/sec	202 MB/sec

Multicore efficiency

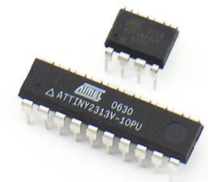


Efficiency on a GPU

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



8-bit processor (Atmel)



- With L=0 (sequential mode), uses less than 1KB RAM.
- 20 MHz clock
- 110 msec/comp. fn for MD6-224 (gcc actual)
- 44 msec/comp. fn for MD6-224 (assembler est.)

Hardware Implementations

FPGA Implementation (MD6-512)

- Xilinx XUP FPGA (14K logic slices)
- 5.3K slices for round-at-a-time
- 7.9K slices for two-rounds-at-a-time
- 100MHz clock
- 240 MB/sec (two-rounds-at-a-time) (Independent of digest size due to memory bottleneck)

Security Analysis

Generate and paste attacks (again)

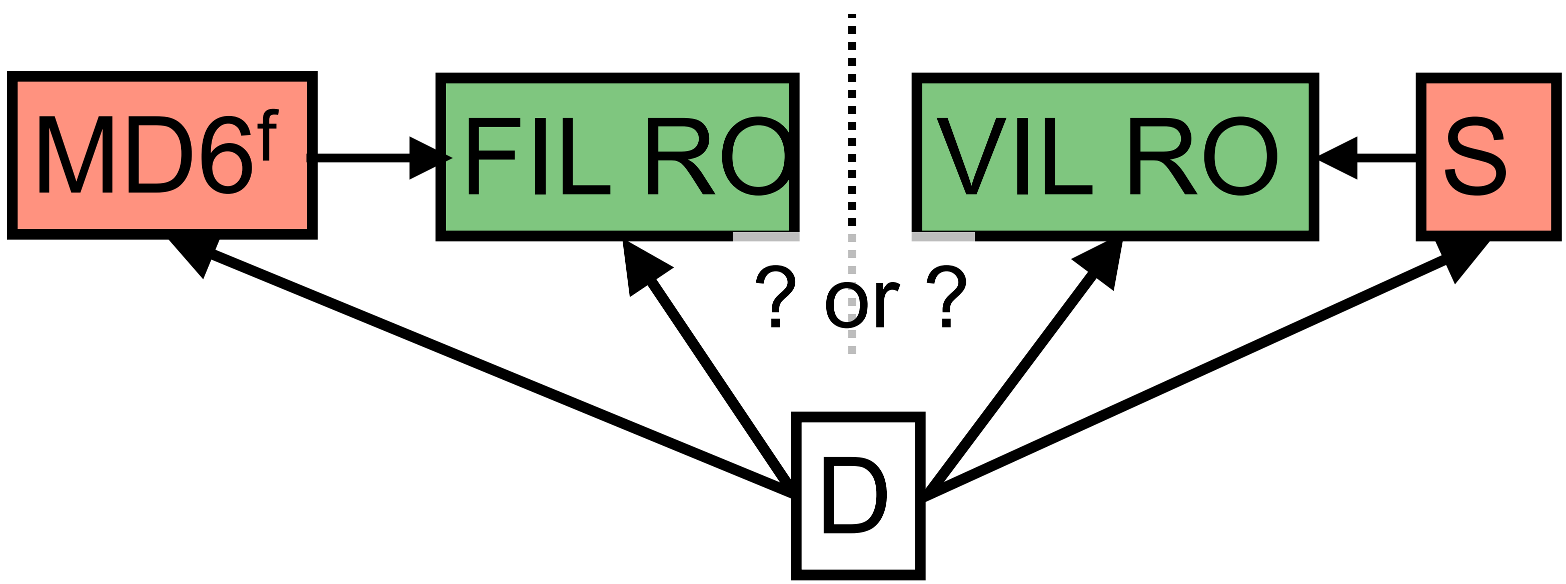
- Because compression functions are “location-aware”, attacks that do speculative computation hoping to “cut and paste it in somewhere” don’t work.

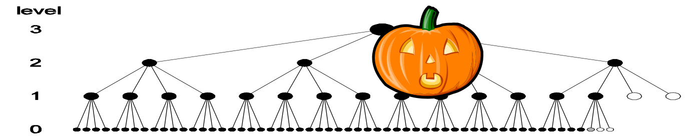
Property-Preservations

- **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 Chris 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).

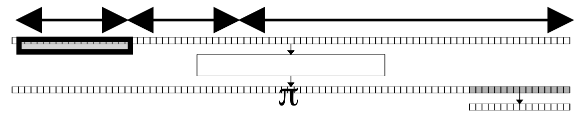




Indifferentiability (I)

- **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…
- **Advantage:** $\epsilon \leq 2\ q^2 / 2^{1024}$
  where q = number of calls (measured in terms of compression function calls).

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.
- **Advantage:** $\epsilon \leq q / 2^{1024} + 2q^2 / 2^{4672}$

SAT-SOLVER attacks

- Code comp. fn. as set of clauses, try to find inverse or collision with Minisat…
- With many days of computing:
  - Solved all problems of 9 rounds or less.
  - Solved some 10- or 11-round ones.
  - Never solved a 12-round problem.
- Note: 11 rounds ≈ 2 rotations (passes over data)

Statistical tests

- Measure influence of an input bit on *all* output bits; use Anderson-Darling A^{*2} test on set of influences.
- Can’t distinguish from random beyond 12 rounds.

Differential attacks don't work

- **Theorem.** *Any s*tandard differential attack has less chance of finding collision than standard birthday attack.
- **Proof.** Determine lower bound on number of active AND gates in 15 rounds using sophisticated backtracking search and days of computing. Derive upper bound on probability of differential path.

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}

Choosing number of rounds

- We don’t know how to break any security properties of MD6 for more than 12 rounds.
- For digest sizes 224 … 512 , MD6 has 80 … 168 rounds.
- Current defaults probably conservative.
- Current choice allows *proof* of resistance to differential cryptanalysis.

Summary

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

THE END

↓

MD6

↓

03744327e1e959fbdcdf7331e959cb2c28101166

Round constants S_i

- Since they only change every 16 steps, let S'_j be the round constant for round j .
- S'_0 = `0x0123456789abcdef`
- $S'_{j+1} = (S'_j \lll 1) \oplus (S'_j \wedge \text{mask})$
- mask = `0x7311c2812425cfa0`