BLAKE
2012 update
Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan
Secure
Secure
Like the 4 others
BLAKE-256, 14 rounds

Hash: 2.5-round preimage \((2^{241})\)
   \(\text{Li, Xu; 2009}\)

Perm: 8-round distinguisher \((2^{242})\)
   \(\text{Biryukov, Nikolic, Roy; 2011}\)
BLAKE-512, 16 rounds

Hash: 2.5-round preimage (2^{481})
   Li, Xu; 2009

Comp: 5-round near-collision (2^{216})
   Su, Wu, Wu, Dong; 2010
Fast
Sandy Bridge Core i3 “bridge” supercop-20120310

Cycles/byte for long messages

- BLAKE-256: 7.49
- BLAKE-512: 5.66
- Skein-512: 6.38
- SHA-256: 17.44
- SHA-512: 11.67
Sandy Bridge Core i3 “bridge” supercop-20120310

Cycles/byte for 64-byte messages

BLAKE-256: 16.38
BLAKE-512: 14.69
Skein-512: 15.50
SHA-256: 46.06
SHA-512: 27.81
Sandy Bridge Core i3 “bridge”
supercop-20120310

Cycles/byte for 8-byte messages

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cycles/byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-256</td>
<td>75.00</td>
</tr>
<tr>
<td>BLAKE-512</td>
<td>119.00</td>
</tr>
<tr>
<td>Skein-512</td>
<td>124.00</td>
</tr>
<tr>
<td>SHA-256</td>
<td>201.50</td>
</tr>
<tr>
<td>SHA-512</td>
<td>224.00</td>
</tr>
</tbody>
</table>
Bulldozer FX-8150 “bulldozer”
supercop-20120310

Cycles/byte for long messages

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cycles/byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-256</td>
<td>11.80</td>
</tr>
<tr>
<td>BLAKE-512</td>
<td>7.00</td>
</tr>
<tr>
<td>Skein-512</td>
<td>8.68</td>
</tr>
<tr>
<td>SHA-256</td>
<td>22.43</td>
</tr>
<tr>
<td>SHA-512</td>
<td>16.15</td>
</tr>
</tbody>
</table>
Bulldozer FX-8150 “bulldozer”
supercop-20120310

Cycles/byte for 64-byte messages

BLAKE-256 | BLAKE-512 | Skein-512 | SHA-256 | SHA-512
---|---|---|---|---
27.97 | 19.88 | 21.94 | 50.88 | 36.53
Bulldozer FX-8150 “bulldozer”
supercop-20120310

Cycles/byte for 8-byte messages

BLAKE-256  134.00
BLAKE-512  164.62
Skein-512  182.38
SHA-256   219.00
SHA-512   291.50
Single-thread treed BLAKE-256 faster than serial

(≈ 25% on “bulldozer”)

Obviously much faster when multi-threaded
Single-thread treed BLAKE-512: < 5.40 cycles/byte on “bulldozer” (7.00 in serial mode)
“the low-end does not go away”

8- and 16-bit MCUs still widely used in automotive and industrial systems

Example: RFID readers, smartmeters
BLAKE: fastest and lowest-memory on ARM cores and MCUs

Faster than SHA-2 on 8- and 16-bit (rotations by 8, 16, 32 help a lot)
Dramatic growth of embedded market
Example: ARM-based SoCs
BLAKE on ARM

Takes advantage of NEON extensions

\[ x4 \text{ speed-up of BLAKE-512 on Cortex A8} \]

Optimizations for non-NEON cores

Rescheduling to exploit op-and-rotate
ARMv7-A Tegra “h5tegra”
supercop-20120310

Cycles/byte for long messages

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cycles/byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE-256</td>
<td>30.10</td>
</tr>
<tr>
<td>BLAKE-512</td>
<td>58.06</td>
</tr>
<tr>
<td>Skein-512</td>
<td>31.36</td>
</tr>
<tr>
<td>SHA-256</td>
<td>28.59</td>
</tr>
<tr>
<td>SHA-512</td>
<td>90.13</td>
</tr>
</tbody>
</table>
“BLAKE is the algorithm with the **highest flexibility**, and the largest number of potential architectures. (...) It is also the only algorithm that has a relatively efficient architecture that is **smaller than the basic iterative architecture of SHA-2**.”

*Homisirikamon, Rogawski, Gaj; CHES 2011*
“the BLAKE family offers one of the best area-time trade-offs and leads to the smallest coprocessors on reconfigurable devices.”

Beuchat, Okamoto, Yamazaki; FPT 2010
Kaps et al. Lightweight implementations of SHA-3 candidates on FPGAs, INDOCRYPT’11

Throughput/area (Mbps)
Xilinx Spartan-3

BLAKE 173.8
Groestl 134.6
Keccak 28.5
JH 34.8
Skein 19.7
ASIC

For optimal efficiency (throughput/area):

- Most scalable
- Most compact

130 nm: $\approx 2$ Gbps, 34 kGE

Guo et al.; DATE 2012
Knežević et al.; IEEE T VLSI, 2011  (Credit: Guo et al.)
Simple
Example: most complex lines in Cantu’s Haskell implementation

\[
a' = a + b + (\text{messageword} \ (i2) \ 'xor' \ \text{constant} \ (i2 + 1))
\]
\[
d' = (d \ 'xor' \ a') \ 'rotateR' \ \text{rot0}
\]
\[
c' = c + d'
\]
\[
b' = (b \ 'xor' \ c') \ 'rotateR' \ \text{rot1}
\]
\[
a'' = a' + b' + (\text{messageword} \ (i2 + 1) \ 'xor' \ \text{constant} \ (i2))
\]
\[
d'' = (d' \ 'xor' \ a'') \ 'rotateR' \ \text{rot2}
\]
\[
c'' = c' + d''
\]
\[
b'' = (b' \ 'xor' \ c'') \ 'rotateR' \ \text{rot3}
\]

https://github.com/killerswan/Haskell-BLAKE
Already many third-party implementations:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Author</th>
<th>Language</th>
<th>Implementation Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM11 asm</td>
<td>Schwabe, Yang, Yang</td>
<td>Java</td>
<td>Pornin</td>
</tr>
<tr>
<td>ARM thumb2</td>
<td>Wenzel-Benner</td>
<td>Java</td>
<td>Greim</td>
</tr>
<tr>
<td>AVR asm</td>
<td>von Maurich</td>
<td>Javascript</td>
<td>Drost</td>
</tr>
<tr>
<td>C (for AVR)</td>
<td>Otte</td>
<td>Matlab</td>
<td>Burgess, Jelley, Smith, Weston</td>
</tr>
<tr>
<td>C (HMAC)</td>
<td>Lazar</td>
<td>Matlab</td>
<td>Steer</td>
</tr>
<tr>
<td>C#</td>
<td>Reichl</td>
<td>Matlab</td>
<td></td>
</tr>
<tr>
<td>CUDA</td>
<td>Bos, Stefan</td>
<td>Perl</td>
<td>Gray</td>
</tr>
<tr>
<td>Go</td>
<td>Chestnykh</td>
<td>PHP</td>
<td>Correa</td>
</tr>
<tr>
<td>Haskell</td>
<td>Cantu</td>
<td>Python</td>
<td>Bugbee</td>
</tr>
</tbody>
</table>

+ eBASH C implementations by Bernstein, Kirst, Leurent, Neves, Pornin, Schwabe

Thanks to all!
Conclusion: BLAKE is...
Secure

with a LARGE margin
Fastest or close-second on desktops & servers
Best performer in embedded systems
Most flexible & scalable in FPGA and ASIC
Implementers-friendly
Thank you!
Thanks D.C. to show support!
(1025 Connecticut Av NW)