

Tiger: a fast hash

Eric Seidel

CS 257: Security Engineering

Overview

- *Hashing: a quick review*
- *Tiger*
- *Why Tiger?*
- *How it works*
- *Q & A*

Hashing: A review

- *Non-linear, (generally) non-reversible function*
- *Purpose: smaller, unique representation of data. (Integrity)*
- *Important to have high avalanche*
- *Important to be fast.*

Tiger

- *Designed for 64-bit processors*
- *Produces larger 192-bit hash (128 compat.)*
- *Designed by Ross Anderson, Eli Biham*
- *Drop-in for MD5*

Why Tiger?

- *Hashing should be fast.*
- *32-bit hashing on 64-bit machine runs at half efficiency.*
- *Uses s-box concept for better non-linearity.*
- *Faster than SHA-1, MD5 on 64-bit*
- *Possibly more secure than SHA-1 or MD5*

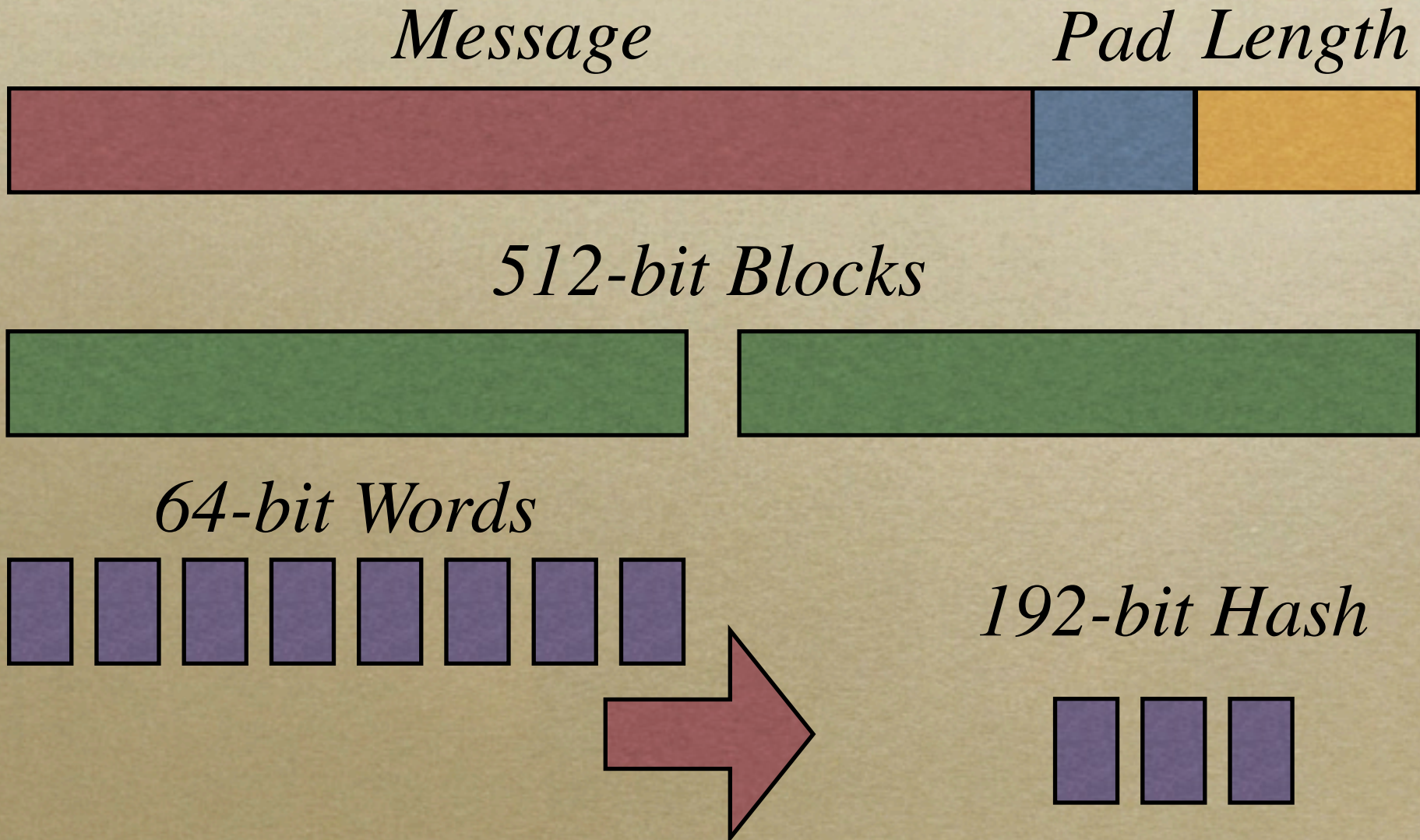
How Tiger Works...



Overview

- *Tiger operates on 512-bit blocks*
- *Each block is broken into 8, 64-bit words*
- *Tiger returns 3 (or 2) 64-bit words*

Overview Diagram



Stages

- *Save ABC*
- *Pass 1, Key Schedule 1*
- *Pass 2, Key Schedule 2*
- *Pass 3*
- *feed-forward ABC*

Save ABC

- *ABC are initially salted with special values.*
- *At the beginning of each successive round ABC are saved for later use with feed-forward.*
- *64_bit_word $aa = a, bb = b, cc = c;$*

Pass 1, detail

- *1 pass = 8 rounds, 1 for each 64-bit word*
- *64-bit words (keys) referred to as $x_0 - x_7$*
- *Each pass uses a multiplier (5,7,9) to redistribute bits between s-box lookups.*
- *$round(a,b,c,x_0, mul);$*

Round function

○ *round(a,b,c,x, multiplier):*

$$c = c \wedge x$$

$$a = a - (s1[c1] \wedge s2[c3] \wedge s3[c5] \wedge s4[c7])$$

$$b = b + (s4[c2] \wedge s3[c4] \wedge s2[c6] \wedge s1[c8])$$

$$b = b * multiplier$$

○ \wedge denotes XOR

S-Boxes

- *s-boxes compose a non-linear function*
- *map from 8 bits into 64.*

- *Available on the author's site:*
<http://www.cs.technion.ac.il/~biham/>

Key Schedule

- *Key-Schedule re-distributes input bits.*
- *Introduces further algorithm complexity.*
- *Rotates words within block.*

- *Each block is only looked at 3 times in the passes, this further distributes the bits.*

Feed-Forward

◦ *Generates new carry values from previous*

◦ $a = a \wedge aa$;

◦ $b = b - bb$;

◦ $c = c + cc$;

Java Demo



Further Thoughts...

A horizontal line of red and black ink scribbles, appearing as if drawn with a marker or brush, extending across the width of the page below the text.

Final thoughts

- *Security*
- *Complexity*
- *Popularity*
- *Performance*

Security

- *Good Avalanche*
- *Long hash (large keyspace)*
- *Little literature*

Complexity

- *More complex than either MD5 or SHA-1*
- *Potential barrier to entry*

Popularity

- *Complexity disadvantage*
- *Relatively new (MDx, and SHA much older)*
- *No pressing need*

Performance

- *Better than SHA-1, MD5 on 64-bit*
- *Comparable on 32-bit*

- *Makes (relatively) large TMTO*
- *Slow (and large!) in hardware*

Q & A

