NIST SHA-3 Submissions:

Grøstl Summary and Security Analysis

Joel Lathrop

Department of Computer Science
Rochester Institute of Technology

January 23, 2009

Outline

1. Grøstl Specification
   - Description
   - Construction
   - Permutation functions
2. Security Analysis
   - Security claims
   - Construction security
   - Resistance against specific attacks
3. Conclusion
   - Conclusions from security analysis
   - A word from my gut

Outline

Description

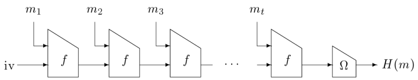
What is Grøstl ?

- A collection of hash functions capable of returning any number of bits in 8-bit steps
    - *n*-bit version called Grøstl -*n*.
- A Merkle-Damgård iterated hash function with a specialized compression function
- Specifics
    - Grøstl is a byte oriented substitution/permutation network
        - S-box is the same as the one used in AES
        - Diffusion technique similar to that used in AES
    - Stated advantage that, since it leverages known good strategies from AES, counter-measures for side-channel attacks have been included
    - Internal state is significantly larger than the output size
    - Not just AES, though. It's a hash function built using the *wide trail strategy*.

Outline

Hash function construction

Figure: The Grøstl hash function.



- Hash function construction
  - Message is padded and split into ℓ-bit blocks, yielding $m_1, m_2, \ldots, m_t$ messages of length ℓ.
  - Padding an N-bit message is done by appending a '1' bit, $-N - 65 \bmod \ell$ '0' bits, and a 64-bit integer of the number of message blocks in the final padded message.
  - These messages are combined sequentially with an ℓ-bit chaining value, initially h_0=iv$_n$, producing an n-bit output.
    - iv is the ℓ-bit represenation of n, (e.g. iv$_{256} = \texttt{00} \ \ldots \ \texttt{01} \ \ \texttt{00}$).
  - $\ell = 512$ bits for $n \leq 256$ and $\ell = 1024$ bits for $n > 256$.

Compression function & Output transformation

- Compression function f:

$$f(h,m) = P(h \oplus m) \oplus Q(m) \oplus h$$

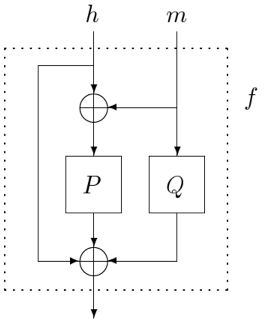
  - P and Q are ℓ-bit permutation functions.

- Output transformation Ω:

$$\Omega(x) = \text{trunc}_n(P(x) \oplus x)$$

  - $\text{trunc}_n(x)$ discards all but the trailing n bits of x.

Figure: The compression function f.



Outline

Basic design

Compression function permutations:

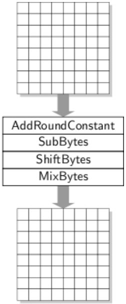
- Two permutations: P and Q
  - Only difference is in the AddRoundConstant step.
- Composed of rounds R, where

  $R = \text{MixBytes} \circ \text{ShiftBytes} \circ \text{SubBytes} \circ \text{AddRoundConstant}$
- Permutations vary based on block size ℓ

	$\ell = 512$	$\ell = 1024$
suggested # rounds	10	14
state size	8x8	8x16

- State is a matrix A mapped from a byte string by consequtively filling columns left to right.

Figure: Round function



Permutation round steps

- AddRoundConstant
  - Given round number i, modifies the state as follows:

$$A \leftarrow A \oplus C[i]$$

where $C[i]$ differs between P and Q and is defined as

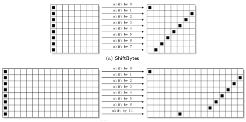
$$C_P[i] = \begin{bmatrix} i & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \end{bmatrix} \text{ and } C_Q[i] = \begin{bmatrix} 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ 00 & 00 & \cdots & 00 \\ i \oplus ff & 00 & \cdots & 00 \end{bmatrix}$$

  - This corresponds to the AddRoundKey step in Rijndael.

Permutation round steps

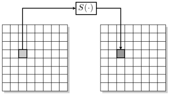
- ShiftBytes and ShiftBytesWide
  - Cyclicly shift all bytes in row *i* of the state matrix σ_i positions to the left.
  - For ShiftBytes, $\sigma = [0, 1, 2, 3, 4, 5, 6, 7]$.
  - For ShiftBytesWide, $\sigma = [0, 1, 2, 3, 4, 5, 6, 11]$.
  - Similar to Rijndael's ShiftRow step, but for a larger internal state.

Figure: ShiftBytes step



- SubBytes
  - Uses the Rijndael S-box to substitute out each byte in A.

Figure: SubBytes step



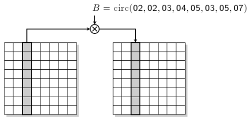
Permutation round steps

- MixBytes
  - Transforms the state matrix A by left-multiplying it by a circulant matrix B, as in $A \leftarrow B \times A$.
  - Multiplication is done in $\mathbb{F}_{256}$ which is defined just as in Rijndael via the irreducible polynomial $x^8 \oplus x^4 \oplus x^3 \oplus x \oplus 1$ over $\mathbb{F}_2$.
  - $B = circ(02, 02, 03, 04, 05, 03, 05, 07)$, which in matrix form is

$$\begin{bmatrix} 02 & 02 & 03 & 04 & 05 & 03 & 05 & 07 \\ 07 & 02 & 02 & 03 & 04 & 05 & 03 & 05 \\ 05 & 07 & 02 & 02 & 03 & 04 & 05 & 03 \\ 03 & 05 & 07 & 02 & 02 & 03 & 04 & 05 \\ 05 & 03 & 05 & 07 & 02 & 02 & 03 & 04 \\ 04 & 05 & 03 & 05 & 07 & 02 & 02 & 03 \\ 03 & 04 & 05 & 03 & 05 & 07 & 02 & 02 \\ 02 & 03 & 04 & 05 & 03 & 05 & 07 & 02 \end{bmatrix}$$

  - Similar to the MixColumn step in Rijndael, except that B is an entirely new matrix derived to work with the *wide trail strategy*.

Figure: MixBytes step



Permutation Design Rational

Design rationale for the round operations

- AddRoundConstant
  - Makes rounds different by simply add round constants to reduce performance penalty.
  - The only transformation where there is a difference between P and Q, so the round constants must differ.
- SubBytes
  - Only nonlinear transformation.
  - Performance: single S-box, no random S-box.
  - S-box from Rijndael, so already well studied and implementation aspects are well understood.
- ShiftBytes and ShiftBytesWide
  - Designed for optimal diffusion.
- MixBytes
  - Designed to support *wide trail strategy*.
  - Based on an error correcting code with the maximum distance separable (MDS) property.

Outline

Security Claims

- Against the hash function as a whole:

Attack type	Claimed complexity	Best known attack
Collision	$2^{n/2}$	$2^{n/2}$
d-collision	$\lg(d) \cdot 2^{n/2}$	$(d!)^{1/d} \cdot 2^{n(d-1)/d}$
Preimage	2^n	2^n
Second Preimage	2^{n-k}	2^n

- Against the compression function:

Attack type	Claimed complexity	Best known attack
Collision	$2^{\ell/4}$	$2^{\ell/3}$
Preimage	$2^{\ell/2}$	$2^{\ell/2}$

Outline

1. Grøstl Specification
   - Description
   - Construction
   - Permutation functions
2. Security Analysis
   - Security claims
   - Construction security
   - Resistance against specific attacks
3. Conclusion
   - Conclusions from security analysis
   - A word from my gut

Construction Security

- Compression Function: $f(h, m) = P(h \oplus m) \oplus Q(m) \oplus h$
  - Proven to be secure if P and Q are ideal [FSZ08].
  - Proof gives limits on evaluations of P and Q.
    - $2^{\ell/4}$ for collision.
    - $2^{\ell/2}$ for pre-image.
  - Since $\ell \geq 2n$, this isn't a problem.
  - Author's don't presume P and Q ideal; simply show construction is.
- Output Transformation: $\Omega(x) = \text{trunc}_n(P(x) \oplus x)$
  - Based on Matyas-Meyer-Oscas construction for an iterated hash function based on a block cipher E_k.

$$g(h, m) = E_h(m) \oplus m$$

  - Consider $g'(m) = E_{h^*}(m) \oplus m$ for some constant h^*.
  - $P(x) \oplus x$ is equivalent to g'.
  - Therefore, they claim Ω is a Matyas-Meyer-Oscas construction and is collision resistance and one-way.

Outline

1. Grøstl Specification
   - Description
   - Construction
   - Permutation functions
2. Security Analysis
   - Security claims
   - Construction security
   - Resistance against specific attacks
3. Conclusion
   - Conclusions from security analysis
   - A word from my gut

Wide trail strategy stuff

A few things relating to *wide trail strategy* [DR02].

Branch numbers

A transformation has a *branch number* B if a difference in $k > 0$ bytes in one column of the input translates to a difference of at least $B - k$ bytes in the output.

Prop ratio approximation

The propogation ratio of a differential/linear trail can be approximated by the prop ratio of its active S-boxes.

Active S-box lower bound

Because Grøstl is structure acourding to the *wide trail strategy*, and ShiftBytes is diffusion optimal, the number of active S-boxes in a four-round trail is lower bounded by B^2 where B is the branch number of the MixBytes transformation.

Differential cryptanalysis

Analysis of a differential attack against P or Q:

- MixBytes has a branch number of 9.
- Maximum difference prop ratio for Grøstl S-box is 2^{-6}.
- $\therefore$ The number of active S-boxes in a 4-round trail is at least 81.
- $\therefore$ There are at least 162 active S-boxes in an 8-round trail.
- $\therefore$ Probability of an 8-round differential trail (presuming independent rounds) is expected to be at most $2^{-6 \cdot 162} = 2^{-972}$.

Linear cryptanalysis

Analysis of a linear attack against P or Q:

- MixBytes has a branch number of 9.
- Maximum correlation for Grøstl S-box is 2^{-3}.
- $\therefore$ The number of active S-boxes in a 4-round trail is at least 81.
- $\therefore$ There are at least 162 active S-boxes in an 8-round trail.
- $\therefore$ Probability of an 8-round linear trail (presuming independent rounds) is expected to be at most $2^{-3 \cdot 162} = 2^{-486}$.

Integral cryptanalysis

- Not yet shown how to apply integral attacks to hash functions, but still might show something about structure.
- Found integral with 2^{170} texts on 7-round Grøstl -256.
  - Balanced on every byte of the input and bit of the output.
  - Similar to those found on AES.
- Found integrals on Grøstl -512 up to 9 rounds.
  - Number of texts is 2^{704}.
  - For 9 rounds, balanced on every byte of the input and bit of the output.

Algebraic cryptanalysis

- From XSL attack [CP02], it's know there are 39 quadratic equations on AES S-box with one more of probability $\frac{255}{256}$.
- Since Grøstl shares S-box with AES, they apply.
- 40 equations $\times$ 200 S-box applications for one AES encryption = 8000 equations on 1600 unknowns.
- There are 1280 S-box applications in compression function of Grøstl -256.
- Grøstl -512 has 3584 S-box applications.
- Authors claim that if Grøstl is broken by an algebraic attack, AES almost certainly will be as well.

Outline

Conclusion

- Grøstl more than just AES wrapped as a hash function
  - It's a hash function built using the *wide trail design strategy*.
- Compression function and output transformation construction seem solid.
- Wide trail strategy seems to offer incredible resistance to differential and linear cryptanalysis.
- Integral attacks don't apply and algebraic attacks aren't proven to be effective yet.
- A cube attack on Grøstl would almost certainly be more difficult than one on AES.

Outline

A word from my gut

Things that make me nervous:

- The round function is invertible.
  - Considering the compression function, it feels like if someone comes up with a decent differential attack, we'll have a pre-image attack, not just a collision attack, staring down our throat.
  - Fortunately, it looks *really hard* to come up with a differential attack.
- It feels that Grøstl , like Rijndael, might have a fairly simple algebraic structure.
  - Not a problem right now, but algebraic cryptanalysis is fairly new and with new discoveries like Dinur and Shamir's *cube attack* [DS08] this simplicity could lead to trouble later on.

Authorship notes

- With Andrew Fitzgerald's gracious permission, this presentation is partially based on and borrows from his previous presentation on CubeHash and Grøstl of December 16, 2008.

- Various diagrams, formulas, and descriptions of Grøstl internals are taken directly from the Grøstl specification [GKM$^+$08].

References

Nicolas Courtois and Josef Pieprzyk.
Cryptanalysis of block ciphers with overdefined systems of equations.
Cryptology ePrint Archive, Report 2002/044, 2002.

Joan Daemen and Vincent Rijmen.
The design of Rijndael: AES — the Advanced Encryption Standard.
Springer-Verlag, 2002.

Itai Dinur and Adi Shamir.
Cube attacks on tweakable black box polynomials.
Cryptology ePrint Archive, Report 2008/385, 2008.

P.A. Fouque, J. Stern, and S. Zimmer.
Cryptanalysis of tweaked versions of SMASH and reparation.
In *Selected Areas in Cryptography, SAC*, 2008.

Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen.
Grøstl – a SHA-3 candidate.
Submission to NIST, 2008.