From: hash-forum@nist.gov on behalf of Martin Schläffer [martin.schlaeffer@iaik.tugraz.at]
Sent: Tuesday, January 25, 2011 3:52 AM
To: Multiple recipients of list
Subject: OFFICIAL COMMENT: Groestl (Round 3)

Dear all,

as announced in December the Grøstl hash function has been tweaked for the final round. The round3mods, updated specification, implementation and cryptanalysis are available at www.groestl.info.

For the final round we have

* significantly increased the size of the round constants to make the internal differential attack and its extensions impossible

* and use different rotation constants in Q to make P and Q more different which further increases the security margin by one round.

Note that the link on the NIST Round3 website still points to the Round2 submission package.

Kind regards,
the Grøstl team

OFFICIAL COMMENT: Grostl (Round 3)

hash-forum@nist.gov [hash-forum@nist.gov] on behalf of Christian Rechberger [c.rechberger@mat.dtu.dk]

Sent: Tuesday, March 20, 2012 2:30 PM

To: HASH-FORUM

Attachments: groestl-brief.pdf (180 KB)

Dear all,
please find a 2-page summary+appendix entitled "Update on Finalist Grøstl" in the attachment.

Best regards
The Grøstl team

Update on Finalist Grøstl

March 20, 2012

Introduction

This note gives the current status of the SHA-3 finalist Grøstl in terms of security and implementation. It is evident that Grøstl is a very strong hash function with a large security margin, despite many cryptanalysis attempts. Grøstl can be faster than SHA-2 on modern high-end CPU architectures with 64-bit or larger register sizes (using AES-specific instructions), provides sufficient performance on 32-bit architectures, is the fastest and also most compact on 8-bit CPUs, and is often top-ranked in all ASIC and FPGA comparisons, especially in resource-constrained settings.

Security

We believe Grøstl offers the best security assurance among all SHA-3 finalists and argue as follows:

- Grøstl and its compression function have received significant formal security analysis in the ideal permutation model. In this model, Grøstl was proved to be indifferentiable from a random oracle up to the birthday bound [1] and the compression function has security bounds against collision, preimage and multi-target preimage attacks [7, 5] matching at least the respective ideal security levels of the hash function.

- Since NIST initiated the SHA-3 competition [25], Grøstl and its building blocks have received the largest amount of cryptanalysis [9, 23, 22, 24, 10, 12, 26, 14, 28, 3, 29, 11, 31, 15, 5] among the finalists.

- The security margin offered by Grøstl was improved from the initial version of the design to the tweaked version proposed in the round 3 of the competition without invalidating most of the cryptanalytic techniques and ideas on the earlier version.

To summarize, the best published cryptanalytic results on the hash function are on 3 rounds for both Grøstl-256 and Grøstl-512 [29], leaving a large security margin for the design, despite a significant cryptanalytic effort. For details, we refer to the Appendix.

Implementation

Here we briefly survey the ranking of Grøstl relative to other SHA-3 finalists in various implementation scenarios.

- Low-cost ASICs: Top 1-2 according to all metrics such as area, or throughput/area [18].
- Low-cost FPGAs: Top 1-2. All surveys of finalists arrive at the same conclusions. Number 1 in [16], number 2 in [17], number 1 in [19].
- Fast hardware: Here the ranking depends on the metric employed. When optimizing for high throughput, Grøstl allows for implementation approaches that achieve high speed that consistently puts it into position 2. This however usually results also in high area requirements, hence by metrics that take area into account put Grøstl in positions 3-5 [13, 20].
- Fast FPGAs: Similarly to the ASIC case, the ranking depends on the metric. Position 2 for high throughput, position 2-4 when area influence is taken into account [8].
- High-end Intel CPUs (and similar architectures) position 3-4 with constant-time implementations. Grøstl-256 outperforms SHA-256, and is even on par with the faster SHA-512 [2]. Grøstl-512 is about 40% slower.
- ARM CPUs with 32-bit architecture: Position 4 according to [30]. Faster implementations are work in progress.
- 8-bit CPUs: Position 1 with respect to all metrics such as ROM, RAM, and speed [6].
- Gains from instruction set extensions: Position 1 according to [4].

Additionally, it seems worth pointing out that Grøstl is the only candidate that allows for significant resource re-use with an AES implementation in software on resource-constraint devices, and especially also in hardware. Finally, we give a number of remarks on side-channel attacks.

- Cost of protection against power and EM side-channel attacks. Unfortunately, very little is known for the SHA-3 finalists, however it is folklore knowledge in the semiconductor industry that SHA-2 family hash functions are more complicated and expensive to protect against those attacks than the AES [27, 21]. Hence, the large body of work on attacks on AES and countermeasures is of great benefit for Grøstl.
- Cost of protection against timing attacks. No additional cost when an AES instruction is available, as on many modern CPUs. Similarly also no additional cost on high-end Intel CPUs (and similar architectures), as the vperm approach is as efficient as the table-based approach. For current ARM based architectures, demonstration of similar advantageous property is work in progress.

References

[1] Elena Andreeva, Bart Mennink, and Bart Preneel. On the Indifferentiability of the Grøstl Hash Function. In Juan A. Garay and Roberto De Prisco, editors, *SCN*, volume 6280 of *LNCS*, pages 88–105. Springer, 2010.

[2] Daniel J. Bernstein and Tanja Lange. The new SHA-3 software shootout. Cryptology ePrint Archive, Report 2012/004, 2012. `http://eprint.iacr.org/`.

[3] Christina Boura, Anne Canteaut, and Christophe De Canniere. Higher-order differential properties of Keccak and Luffa. In *Fast software encryption*, volume 6733 of *LNCS*, pages 252–269. Springer, 2011.

[4] Jeremy Constantin, Andreas Burg, and Frank K. Gurkaynak. Investigating the Potential of Custom Instruction Set Extensions for SHA-3 Candidates on a 16-bit Microcontroller Architecture. Cryptology ePrint Archive, Report 2012/050, 2012. `http://eprint.iacr.org/`.

[5] Sareh Emami, Praveen Gauravaram, Josef Pieprzyk, and Ron Steinfeld. (Chosen-multi-target) preimage attacks on reduced Grøstl-0. Draft paper, 2012. Available at `http://web.science.mq.edu.au/~rons/pubs.html`.

[6] Johannes Feichtner. Effcient Grøstl-256 Implementations for the AVR 8-bit Microcontroller Architecture, 2011. `http://www.groestl.info/groestl-avr8asm.pdf`.

[7] Pierre-Alain Fouque, Jacques Stern, and Sébastien Zimmer. Cryptanalysis of Tweaked Versions of SMASH and Reparation. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *LNCS*, pages 136–150. Springer, 2008.

[8] Kris Gaj, Ekawat Homsirikamol, Marcin Rogawski, Rabia Shahid, , and Malik Umar Sharif. Comprehensive Evaluation of High-Speed and Medium-Speed Implementations of Five SHA-3 Finalists Using Xilinx and Altera FPGAs. The Third SHA-3 Candidate Conference, 2012.

[9] 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's SHA-3 Cryptographic Hash Function Competition, 2008. Available at `http://www.groestl.info/specification.html`.

[10] Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Grøstl – A SHA-3 Candidate. Second Round of NIST's SHA-3 Competition, 2009. Available at `http://www.groestl.info/`.

[11] Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Grøstl – A SHA-3 Candidate. A Finalist of NIST's SHA-3 Cryptographic Hash Function Competition, 2011. Available at `http://www.groestl.info/`.

[12] Henri Gilbert and Thomas Peyrin. Super-Sbox cryptanalysis: improved attacks for AES-like permutations. In *Fast software encryption*, LNCS, pages 365–383. Springer, 2010.

[13] Frank K. Gurkaynak, Kris Gaj, Beat Muheim, Ekawat Homsirikamol, Christoph Keller, Marcin Rogawski, Hubert Kaeslin, and Jens-Peter Kaps. Lessons Learned from Designing a 65 nm ASIC for Evaluating Third Round SHA-3 Candidates. The Third SHA-3 Candidate Conference, 2012.

[14] Kota Ideguchi, Elmar Tischhauser, and Bart Preneel. Improved Collision Attacks on the Reduced-Round Grøstl Hash Function. In Mike Burmester, Gene Tsudik, Spyros S. Magliveras, and Ivana Ilic, editors, *ISC*, volume 6531 of *LNCS*, pages 1–16. Springer, 2010.

[15] Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Improved Rebound Attack on the Finalist Grøstl. LNCS. Springer, 2012. To appear.

[16] Bernhard Jungk and Jürgen Apfelbeck. Area-Efficient FPGA Implementations of the SHA-3 Finalists. In Peter M. Athanas, Jürgen Becker, and René Cumplido, editors, *ReConFig*, pages 235–241. IEEE Computer Society, 2011.

[17] Jens-Peter Kaps, Panasayya Yalla, Kishore Kumar Surapathi, Bilal Habib, Susheel Vadlamudi, Smriti Gurung, and John Pham. Lightweight Implementations of SHA-3 Candidates on FPGAs. In Daniel J. Bernstein and Sanjit Chatterjee, editors, *INDOCRYPT*, volume 7107 of *LNCS*, pages 270–289. Springer, 2011.

[18] Elif Bilge Kavun and Tolga Yalcin. On the Suitability of SHA-3 Finalists for Lightweight Applications. The Third SHA-3 Candidate Conference, 2012.

[19] Stéphanie Kerckhof, François Durvaux, Nicolas Veyrat-Charvillon, Francesco Regazzoni, Guerric Meurice de Dormale, and François-Xavier Standaert. Compact FPGA Implementations of the Five SHA-3 Finalists. In Emmanuel Prouff, editor, *CARDIS*, volume 7079 of *LNCS*, pages 217–233. Springer, 2011.

[20] Kashif Latif, M Muzaffar Rao, Arshad Aziz, and Athar Mahboob. Efficient Hardware Implementations and Hardware Performance Evaluation of SHA-3 Finalists. The Third SHA-3 Candidate Conference, 2012.

[21] Stefan Mangard. Personal communcation, 2011.

[22] Florian Mendel, Thomas Peyrin, Christian Rechberger, and Martin Schläffer. Improved Cryptanalysis of the Reduced Grøstl Compression Function, ECHO Permutation and AES Block Cipher. In Michael J. Jacobson Jr. and Vincent Rijmen and Reihaneh Safavi-Naini, editor, *Selected Areas in Cryptography*, volume 5867 of *LNCS*, pages 16–35. Springer, 2009.

[23] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl. In Orr Dunkelman, editor, *Fast Software Encryption*, volume 5665 of *LNCS*, pages 260–276. Springer, 2009.

[24] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Rebound Attacks on the Reduced Grøstl Hash Function. In Josef Pieprzyk, editor, *The Cryptographers' Track at the RSA Conference(CT-RSA)*, volume 5985 of *LNCS*, pages 350–365. Springer, 2010.

[25] NIST. Announcing the Development of New Hash Algorithms for the Revision of Federal Information Processing Standard (FIPS) 180-2, Secure Hash Standard, January 2007. This notice by NIST is available at `http://www.csrc.nist.gov/pki/HashWorkshop/timeline.html` with the Docket No: 061213336-6336-01.

[26] Thomas Peyrin. Improved Differential Attacks for ECHO and Grøstl. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 370–392. Springer, 2010.

[27] Emmanuel Prouff. Use of Hash Functions in the Smart Card Industry. Ecrypt II Hash Workshop, 2011. `http://www.ecrypt.eu.org/hash2011/program.shtml`.

[28] Yu Sasaki, Yang Li, Lei Wang, Kazuo Sakiyama, and Kazuo Ohta. Non-full-active Super-Sbox Analysis: Applications to ECHO and Grøstl. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 38–55. Springer, 2010.

[29] Martin Schläffer. Update Differential Analysis of Grøstl, 2011. Available at `http://www.groestl.info/analysis.html`.

[30] Christian Wenzel-Benner, Jens Gräf, John Pham, and Jens-Peter Kaps. XBX Benchmarking Results January 2012. The Third SHA-3 Candidate Conference, 2012.

[31] Shuang Wu, Dengguo Feng, Wenling Wu, Jian Guo, Le Dong, and Jian Zou. (Pseudo) Preimage Attack on Round-Reduced Grøstl Hash Function and Others. Fast Software Encryption, 2012. To appear.

A Analysis of Grøstl

A good amount of analysis has been carried out on Grøstl since its submission to the SHA-3 competition from the perspective of both cryptanalysis and formal analysis. A lot of this analysis was done by the design team, and this analysis was initiated before the submission to the SHA-3 competition. Several improvements to the analysis have been made since then, but these have for the most part consisted in finding ways of exploiting more available degrees of freedom. As a result, the best current attacks on round-reduced Grøstl leave only few remaining degrees of freedom for the attacker. Nevertheless, we decided to slightly tweak Grøstl before round 3 of the SHA-3 competition to increase its security margin. While the tweak does not affect performance in any significant way, it does render the internal differential attack [26] and all its extensions infeasible. Furthermore, the tweak also decreases the efficiency of the rebound attack on both the hash and compression function by one round. Note that the permutation results were not affected by the change and all other cryptanalysis results can easily be adapted to tweaked Grøstl [29]. In the following, we give an short overview of the best known attacks on Grøstl as well as its formal analysis.

A.1 Formal analysis

Andreeva, Mennink and Preneel [1] proved that in the ideal permutation model all versions of Grøstl are indifferentiable from a random oracle up to the birthday bound. Fouque *et al.* [7] used ideal permutation model to establish security bounds for the Grøstl compression function against collision and preimage attacks and they apply to hash function without output transformation. Recently, Emami *et al.* [5] extended the analysis of Fouque *et al.* [7] to derive bounds in the multi-target preimage attack setting. These security bounds for the compression function match at least the ideal security levels of the hash function.

A.2 Hash Function Analysis

The best published cryptanalytic results on the hash function are on 3 rounds for both Grøstl-256 and Grøstl-512 by Schläffer [29] based on the rebound attack [23] which was invented during the design of Grøstl. We believe that it might be possible to extend the attack by one or even two rounds in the future, but then no degrees of freedom are available to the attacker anymore to extend the attack to more rounds. This is also supported by the analysis of the compression function which suggests that Grøstl offers a large security margin. Even if the adversary has full access to the wide-pipe chaining value no collision or preimage attacks could be found.

Table 1: Summary of analysis the Grøstl hash functions.

Target	Rounds	Time	Memory	Type	Reference
Grøstl-256	3/10	2^{64}	-	collision	[29]
Grøstl-512	3/14	2^{192}	-	collision	[29]

A.3 Hash Function Analysis with Access to the Chaining Input

Grøstl is among the very few SHA-3 candidates with security claims that go beyond those required by NIST. The Grøstl compression function is claimed to be collision resistance and preimage resistance up to the level needed for the hash function. Since the chaining input of Grøstl is as big as the message input, this increases the degrees of freedom of an attack significantly. Even in this much simpler setting no collision or preimage attack is found. This clearly serves as a reassurance of the collision and preimage resistance of the Grøstl hash function. In this simple setting, pseudo-collisions for Grøstl-256 on 6/10 rounds and Grøstl-512 on 6/14 rounds have been shown in in [29]. Furthermore, Wu *et al.* [31] presented pseudo-preimage attacks on 5/10 rounds of Grøstl-256 and 8/14 rounds of Grøstl-512. However, the memory requirements of the attacks are quite high.

Table 2: Summary of analysis for Grøstl when the adversary has access to the chaining input.

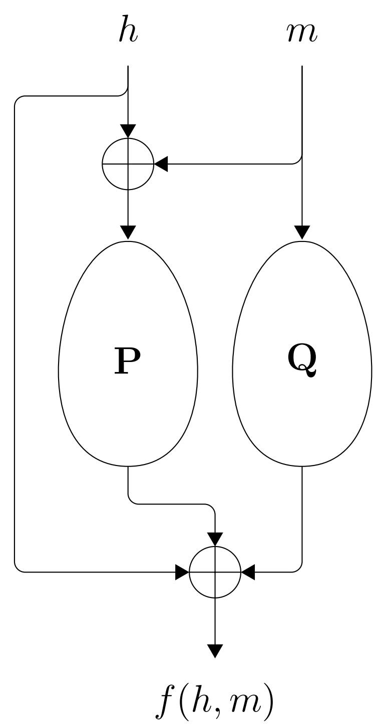
Target	Rounds	Time	Memory	Type	Reference
Grøstl-256	6/10	2^{120}	2^{64}	pseudo-collision	[29]
	5/10	$2^{244.85}$	$2^{230.13}$	pseudo-preimage	[31]
Grøstl-512	6/14	2^{180}	2^{64}	pseudo-collision	[29]
	8/14	$2^{507.32}$	2^{507}	pseudo-preimage	[31]

A.4 Non-random properties of building blocks

Non-random properties of the Grøstl hash function are not known. Here, we consider non-random properties of some of the underlying building blocks. The Grøstl compression function is claimed to be collision and preimage resistant up to the level needed for the hash function. Even though these properties are not strictly necessary, they serve as a reassurance of the collision and preimage resistance of the Grøstl hash function. On the other hand, the Grøstl compression function is known to have some non-random properties independent of the permutations although they do not contradict the security against collision and (second) preimage attacks for the compression function. Hence, the wide pipe and the strong output transformation are essential parts of the design. Nevertheless, here we give an incomplete list of known non-random properties of the compression function.

- Many fixed points can be found for the compression function in time 1 [11].
- Distinguishers based on k-sums (of value zero) or differential q-multicollisions are easy to find for the compression function. We give one example for a 4-sum here: Let $H_1 + H_2 + H_3 + H_4 = 0$ and $H_1 + H_2 = M_1 + M_2$, then $f(H_1, M_1) + f(H_2, M_2) + f(H_3, M_1) + f(H_4, M_2) = 0$. Note that this also implies $H_1 + H_2 = H_3 + H_4 = \Delta_1$ and $f(H_1, M_1) + f(H_2, M_2) = f(H_3, M_1) + f(H_4, M_2) = \Delta_2$.
- Generalized birthday collision attack in time 2^{171} for Grøstl-256 and 2^{341} for Grøstl-512 [11].
- Memoryless preimage attack in time $2^{b/2}$, where $b \geq 2n$ is the output size of the compression function. Note that for a given target T, one can compute M, X using cycle finding algorithms such that $T = H + P(H + M) + Q(M) = X + P(X) + M + Q(M)$ with $H = X + M$.

Differential distinguishers for the Grøstl-256 and Grøstl-512 permutations have been published in [28, 15, 12]. The best ones (in number of rounds) are for 9/10 rounds of Grøstl-256 in time 2^{368} and for 10/14 rounds of Grøstl-512 [15] in time 2^{392}. Moreover, Boura *et al.* have shown non-random properties for the Grøstl-256 permutations in time 2^{509}[3]. However, we want to remark that although the complexities of these distinguishers are less than those on ideal permutations, these complexities are often far above the claimed security levels of the hash function.



Caswell, Sara J.

From: Sami Anand <sam.anand1305@gmail.com>
Sent: Thursday, May 24, 2012 1:30 PM
To: internal-hash
Cc: HASH-FORUM
Subject: OFFICIAL COMMENT: Grøstl (Round 3)
Attachments: Average Speed of Grøstl on Processors.docx

Follow Up Flag: Follow up
Flag Status: Flagged

Respected,

I have attached a file with the results calculated on ARM processor platforms using IAR embedded workbench for ARM .

--
Regards
Sami Anand
+91-9878581768

Average Speed of Grøstl on Processors

Processor Name	Speed(Cycles/byte)
Cortex A9	154.166
Cortex M3	129.69
ARM7TDMI	254.297

Table 1: Average Speed Grøstl on Processors