Grøstl Addendum

Gauravaram, Praveen; Knudsen, Lars Ramkilde; Matusiewicz , Krystian; Mendel, Florian; Rechberger, Christian; Schläffer, Martin; Thomsen, Søren Steffen; Archives, The Pennsylvania State University CiteSeerX

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Grøstl Addendum

Praveen Gauravaram Lars R. Knudsen Krystian Matusiewicz Florian Mendel
Christian Rechberger Martin Schläffer Søren S. Thomsen

September 15, 2009

1 Introduction

This document is an addendum to the submission document of Grøstl, which was selected for the second round of NIST's SHA-3 competition [18]. We stress that we do **not** change the specification of Grøstl. In other words, Grøstl is defined exactly as specified in the original submission document [8]. In this document we mention a few *alternative* descriptions of our SHA-3 candidate Grøstl and describe recent analysis results on Grøstl.

We briefly recall that the Grøstl compression function is based on two large and distinct ℓ-bit permutations P and Q (where $\ell \geq 2n$, n being the output size of the hash function), and is defined as $f(h, m) = P(h \oplus m) \oplus Q(m) \oplus h$, where h is the chaining value and m is the message block. The permutations P and Q are built using the wide trail design strategy. A Merkle-Damgård iteration [6, 17] of the compression function is applied, and it is followed by an output transformation defined as $\omega(x) = \mathsf{trunc}_n(P(x) \oplus x)$, where trunc_n indicates truncation to n bits. When $n \leq 256$, we have $\ell = 512$, and when $n > 256$ we have $\ell = 1024$.

2 Analysis results

In this section, we describe the current state of the art with respect to the analysis of Grøstl.

2.1 Rebound attacks

Recently, a new attack method for the cryptanalysis of hash functions has been proposed: the rebound attack [16]. It gives the best known results for a number of AES-based hash functions [12], including many SHA-3 candidates [13–15, 21]. In general, the rebound attack works with any differential or truncated differential. However, the diffusion properties of AES based hash functions allow a very simple construction of good truncated differential paths, which facilitates the analysis.

The rebound attack is most successful if a high number of degrees of freedom is available. Therefore, attacks on hash functions with a key schedule to the underlying block cipher or other sources of freedom are more likely to succeed (see the recent attacks on Whirlpool [12] or LANE [13]). However, Grøstl has been designed to limit the degrees of freedom available to an attacker. Moreover, in attacks on the *hash* function, much fewer degrees of freedom are available (compared to an attack on the compression function). As shown in Table 1, the best attacks on the hash function for Grøstl-256 and Grøstl-512 are for 4 and 5 rounds (out of 10 and 14), respectively.

On the other hand, in Table 2 we show recent results for the Grøstl compression function. The best (collision) attack on the compression function is for 7 rounds of Grøstl-256 and Grøstl-512. An extension of these attacks to more rounds seems unlikely, since the remaining degrees of freedom in the attacks are close to zero. Therefore, Grøstl still enjoys a comfortable security margin.

Table 1: Summary of rebound analysis for the round-reduced Grøstl hash functions.

Target	Rounds	Time	Memory	Type	Reference
Grøstl-256	3/10	2^{64}	-	collision	[9]
hash function	4/10	2^{64}	2^{64}	collision	[9]
Grøstl-512	4/14	2^{64}	2^{64}	collision	[9]
hash function	5/14	2^{176}	2^{64}	collision	[9]

Table 2: Summary of rebound analysis for the round-reduced Grøstl compression function. Semi-free-start collision is a collision in the compression function of the form $(h, m), (h, m^*)$, where $m \neq m^*$.

Target	Rounds	Time	Memory	Type	Reference
Grøstl-256 compression function	5/10	2^{64}	2^{64}	semi-free-start collision	[16]
	6/10	2^{64}	2^{64}	semi-free-start collision	[14]
	7/10	2^{120}	2^{64}	semi-free-start collision	[9]
Grøstl-512	6/14	2^{96}	2^{64}	semi-free-start collision	[9]
compression function	7/14	2^{152}	2^{64}	semi-free-start collision	[9]

2.2 Kelsey's observations

Kelsey [11] noted that without truncation, the Grøstl hash function does not protect against length extension attacks, and he argues that the "$P(x) \oplus x$" part of the output transformation therefore accomplishes little security. In the following, as well as in Section 3.1, we argue why the "$P(x) \oplus x$" part in the output transformation still serves an important purpose.

If the output of the last iteration of the compression function is merely truncated to form the output of the hash function, then Wagner's generalized birthday attack [20] on the compression function would extend to the hash function, and it would have a complexity of $2^{n/3}$ since it can be applied to the truncated (n-bit) hash value. With the "$P(x) \oplus x$" part, Wagner's generalized birthday attack has to be applied on an internal ℓ-bit value, and since $\ell \geq 2n$, the attack has complexity above the birthday attack on the hash function.

2.3 Consideration of recent attacks on AES

Recently, a number of surprisingly effective cryptanalytic results on AES-256 and AES-192 have been published [2–4]. These results exploit non-ideal properties of the AES-256 and AES-192 key schedules. Grøstl has no key schedule, and therefore the attacks are of no relevance to Grøstl. In fact, Grøstl was designed to be permutation based to allow for simple analysis, and to entirely avoid attacks mounted on the key schedule. When a hash function is based on a block cipher, such attacks are often difficult to mount, but also difficult to exclude.

3 Alternative descriptions of Grøstl

Alternative descriptions of a function serve several purposes. They potentially bring greater insights into its security, and may also lead to more efficient implementations. In the standard description of Grøstl, the hash function iterates a permutation-based compression function, and then applies an output transformation to form the final hash of a message. However, as we shall see in this section, there are other ways of describing Grøstl.

3.1 The output transformation as a compression function call

The output transformation is defined as $\omega(x) = \mathsf{trunc}_n(P(x) \oplus x)$. Notice that $\omega(x) = \mathsf{trunc}_n(f(x, 0^\ell) \oplus Q(0^\ell))$. Hence, if H is the `Grøstl` hash function, $\tilde{H}$ is `Grøstl` *without the output transformation* and M is the already padded message, then $H(M) = \mathsf{trunc}_n(\tilde{H}(M \| 0^\ell) \oplus Q(0^\ell))$, which is also illustrated by Figure 1. Since the XOR with $Q(0^\ell)$ has no cryptographic significance, we may ignore it and consider the description $\mathsf{trunc}_n(\tilde{H}(M \| 0^\ell))$. The suffix 0^ℓ can be seen as an additional padding block.

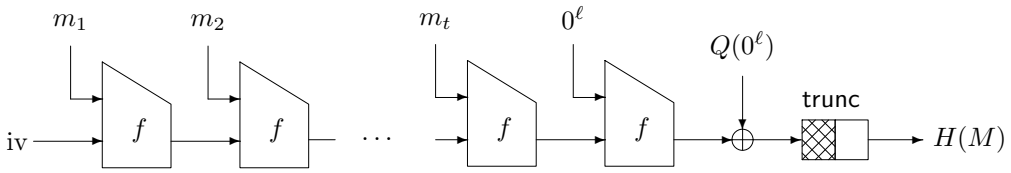

Figure 1: An alternative description of the `Grøstl` hash function.

This description more clearly shows the validity of John Kelsey's observation on the output transformation. What precludes an attack based on this observation is the truncation from ℓ to n bits. Since at least n bits are dropped in this truncation, the probability of correctly guessing those bits is about 2^{-n}. The alternative description can also be seen as an indication that `Grøstl` is in fact an instance of the chop-MD construction, which prevents extension attacks [5].

Some implementations of `Grøstl` might benefit from this alternative description. It shows that one does not have to specifically implement an output transformation function; the compression function can be used instead. Although this is not likely to improve the speed of implementations, it might reduce code size or area.

Finally, the alternative description shows that the "$P(x) \oplus x$" part of the output transformation does not have unexpected negative side effects. Hence, it does not lead to attacks that would not be possible with mere truncation. Since, as mentioned in Section 2.2, mere truncation leads to attacks that are not possible with the true definition of the `Grøstl` output transformation, we can conclude that the "$P(x) \oplus x$" part strictly improves the security of the hash function.

3.2 Tessaro's observation

Similar to the above description of `Grøstl`, Stefano Tessaro [19] observed that $H(M) = \mathsf{trunc}_n(\hat{H}(M \| Q^{-1}(0^\ell)) \oplus Q^{-1}(0^\ell))$, where $\hat{H}$ is the MDP iteration [10] of `Grøstl`'s compression function, with permutation $\pi(x) = x \oplus Q^{-1}(0^\ell)$.

3.3 Barreto's observation

Paulo Barreto observed [1] that the `Grøstl` compression function can be seen as an Even-Mansour cipher [7] in Davies-Meyer mode, which is defined as $f(h, m) = E_m(h) \oplus h$ for a block cipher E keyed via m. In the case of `Grøstl`, the block cipher is defined as $E_k(x) = P(k \oplus x) \oplus Q(k)$, where Q can be seen as a key schedule. In other words, the key is XORed with the plaintext (pre-whitening), the resulting value is permuted, and the output is XORed with a permuted version of the key (post-whitening).

4 Conclusion

A good amount of analysis has been carried out on `Grøstl` since its submission to the SHA-3 competition. Most of this analysis was done by the design team, and this analysis was initiated before

the submission. Some 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.

External (as well as internal) analysis has provided alternative descriptions of the `Grøstl` hash function and the `Grøstl` compression function. Such alternative descriptions might provide new insights into the security of the hash function (e.g., new security proofs) and improved implementations in some settings.

Finally, we acknowledge John Kelsey's observations on the role of $P(x) \oplus x$ in the output transformation, but we point out that the output transformation serves other important purposes than merely protection against length extension attacks.

Acknowledgements

We would like to thank all the people that have contributed to the analysis of `Grøstl`.

References

[1] P. S. L. M. Barreto. An observation on Grøstl. Comment submitted to the NIST hash function mailing list, `hash-forum@nist.gov`. Available: `http://www.larc.usp.br/~pbarreto/Grizzly.pdf` (2009/08/07), 2008.

[2] A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir. Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds. Cryptology ePrint Archive, Report 2009/374, 2009. `http://eprint.iacr.org/`.

[3] A. Biryukov and D. Khovratovich. Related-key Cryptanalysis of the Full AES-192 and AES-256. In M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009, Proceedings*, Lecture Notes in Computer Science. Springer, 2009. To appear.

[4] A. Biryukov, D. Khovratovich, and I. Nikolić. Distinguisher and Related-Key Attack on the Full AES-256. In S. Halevi, editor, *Advances in Cryptology – CRYPTO 2009, Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249. Springer, 2009.

[5] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448. Springer, 2005.

[6] I. Damgård. A Design Principle for Hash Functions. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1990.

[7] S. Even and Y. Mansour. A Construction of a Cipher from a Single Pseudorandom Permutation. *Journal of Cryptology*, 10(3):151–162, 1997.

[8] P. Gauravaram, L. R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen. Grøstl – a SHA-3 candidate. Submission to NIST, 2008. Available online at `http://www.groestl.info`.

[9] P. Gauravaram, L. R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen. Cryptanalysis Results on `Grøstl`, 2009. In preparation.

[10] S. Hirose, J. H. Park, and A. Yun. A Simple Variant of the Merkle-Damgård Scheme with a Permutation. In K. Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 113–129. Springer, 2007.

[11] J. Kelsey. Some notes on Grøstl. Comment submitted to the NIST hash function mailing list, `hash-forum@nist.gov`. Available: `http://ehash.iaik.tugraz.at/uploads/d/d0/Grostl-comment-april28.pdf` (2009/08/07), 2009.

[12] M. Lamberger, F. Mendel, C. Rechberger, V. Rijmen, and M. Schläffer. Rebound Distinguishers: Results on the Full Whirlpool Compression Function. In M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009, Proceedings*, Lecture Notes in Computer Science. Springer, 2009. To appear.

[13] K. Matusiewicz, M. Naya-Plasencia, I. Nikolić, Y. Sasaki, and M. Schläffer. Rebound Attack on the Full LANE Compression Function. In M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009, Proceedings*, Lecture Notes in Computer Science. Springer, 2009. To appear.

[14] F. Mendel, T. Peyrin, C. Rechberger, and M. Schläffer. Improved Cryptanalysis of the Reduced Grøstl Compression Function, ECHO Permutation and AES Block Cipher. In M. J. Jacobson, V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography 2009, Proceedings*, Lecture Notes in Computer Science. Springer, 2009. To appear.

[15] F. Mendel, C. Rechberger, and M. Schläffer. Cryptanalysis of Twister. In M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, editors, *Applied Cryptography and Network Security 2009, Proceedings*, volume 5536 of *Lecture Notes in Computer Science*, pages 342–353. Springer, 2009.

[16] F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen. The Rebound Attack: Cryptanalysis of Reduced Whirlpool and `Grøstl`. In O. Dunkelman, editor, *Fast Software Encryption 2009, Proceedings*, volume 5665 of *Lecture Notes in Computer Science*, pages 260–276. Springer, 2009.

[17] R. C. Merkle. One Way Hash Functions and DES. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1990.

[18] National Institute of Standards and Technology. The SHA-3 competition website. Available: `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html` (2009/08/26).

[19] S. Tessaro. Personal communication, August 2009.

[20] D. Wagner. A Generalized Birthday Problem. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–303. Springer, 2002.

[21] S. Wu, D. Feng, and W. Wu. Cryptanalysis of the LANE Hash Function. In M. J. Jacobson, V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography 2009, Proceedings*, Lecture Notes in Computer Science. Springer, 2009. To appear.