Tweaks on Grøstl

http://www.groestl.info

Praveen Gauravaram[1], Lars R. Knudsen[1], Krystian Matusiewicz[2], Florian Mendel[3], Christian Rechberger[4], Martin Schläffer[3], and Søren S. Thomsen[1]

[1]Department of Mathematics, Technical University of Denmark, Matematiktorvet 303S, DK-2800 Kgs. Lyngby, Denmark
[2]Intel Technology Poland, Juliusza Słowackiego 173, 80-298 Gdansk, Poland
[3]Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Inffeldgasse 16a, A-8010 Graz, Austria
[4]Department of Electrical Engineering ESAT/COSIC, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium

January 16, 2011

1 Introduction

Grøstl is a candidate in the SHA-3 hash function competition, and it was recently selected as one of the five finalists in the competition. Upon this selection, we have decided to make a few simple changes to the constants of the hash algorithm in order to increase its security margin without penalizing its speed. This document describes the changes. From now on, the original version of Grøstl shall be called Grøstl-0. In other words, Grøstl refers to the tweaked version.

In this note, we reuse the notation of the original submission document [1]. We recall here, however, that the Grøstl compression function is defined as

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

where P and Q are two different permutations. For the small variants of Grøstl (returning a hash value of length up to 256 bits), we assign the index 512 to these permutations (since they operate on 512-bit values), and for the large variants (returning hash values from 264 up to 512 bits), we assign the index 1024.

2 Tweaks

The general purpose of the tweaks is to make the permutations P and Q used inside the Grøstl compression function more different. This is achieved in two manners:

1. by changing the shift values used in the permutation Q, and

2. by using "bigger" round constants in P and Q.

We describe the changes on the 512-bit permutations and on the 1024-bit permutations separately.

2.1 512-bit permutations (`Grøstl`-224/256)

Recall that the 512-bit permutations are used in all `Grøstl` variants returning up to 256 bits. Since output sizes of 224 and 256 bits are the most relevant ones, we collectively denote all these `Grøstl` variants by `Grøstl`-224/256.

2.1.1 New Shift Values

In the original version `Grøstl`-0-224/256, the transformation ShiftBytes was used in both P_{512} and Q_{512}. In the new version of `Grøstl`, we introduce different ShiftBytes values to be used in Q_{512}. The ShiftBytes values in P_{512} are kept the same. ShiftBytes moves all bytes in row i of the state matrix by σ_i positions to the left. Then, the new vector σ of Q_{512} is given as follows (`Grøstl`-0: $\sigma = [0, 1, 2, 3, 4, 5, 6, 7]$):

$$Q_{512}: \quad \sigma = [1, 3, 5, 7, 0, 2, 4, 6]$$

These new shift values are also shown in Figure 1.

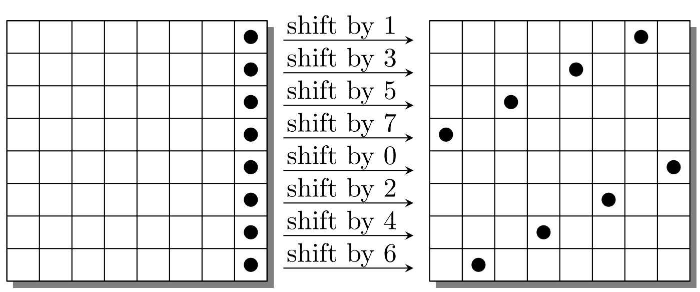

Figure 1: The transformation ShiftBytes of permutation Q_{512}.

2.1.2 New Round Constants

In the original version `Grøstl`-0-224/256, the round constants $C[i]$ of P_{512} and Q_{512} used in the transformation AddRoundConstant in round i were sparse with only a single byte value different from zero.

In the tweaked `Grøstl`-224/256, we increase the round constants to the following values:

$$P_{512}: \; C[i] = \begin{bmatrix} 00 \oplus i & 10 \oplus i & 20 \oplus i & 30 \oplus i & 40 \oplus i & 50 \oplus i & 60 \oplus i & 70 \oplus i \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

and

$$Q_{512}: \ C[i] = \begin{bmatrix} \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} \\ \mathsf{ff} \oplus i & \mathsf{ef} \oplus i & \mathsf{df} \oplus i & \mathsf{cf} \oplus i & \mathsf{bf} \oplus i & \mathsf{af} \oplus i & \mathsf{9f} \oplus i & \mathsf{8f} \oplus i \end{bmatrix}$$

where i is the round number viewed as an 8-bit value, and all other values are written in hexadecimal notation.

We note that the XOR with ff in Q_{512} can be viewed as the application of a different S-box, and hence it can be implemented as a new S-box table if desired.

2.2 1024-bit permutations (Grøstl-384/512)

Recall that the 1024-bit permutations are used in all Grøstl variants returning from 264 up to 512 bits. Since output sizes of 384 and 512 bits are the most relevant ones, we collectively denote all these Grøstl variants by Grøstl-384/512.

2.2.1 New Shift Values

In the original version Grøstl-0-384/512, the same transformation ShiftBytesWide was used in both permutations P_{1024} and Q_{1024}. In the new version of Grøstl, we use different constant values in the ShiftBytesWide transformation of permutation Q_{1024}. The ShiftBytesWide values in P_{512} are kept the same. The new vector σ of Q_{1024} is given as follows (Grøstl-0: $\sigma = [0, 1, 2, 3, 4, 5, 6, 11]$):

$$Q_{1024}: \qquad \sigma = [1, 3, 5, 11, 0, 2, 4, 6]$$

These new shift values are also shown in Figure 2.

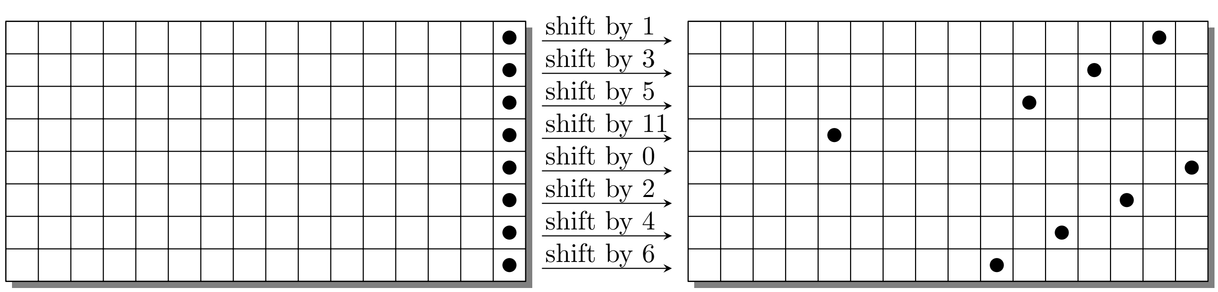

Figure 2: The $\mathsf{ShiftBytesWide}_Q$ transformation.

2.2.2 New Round Constants

In the original version Grøstl-0-384/512, the round constants $C[i]$ of P_{1024} and Q_{1024} used in the transformation AddRoundConstant in round i were sparse with only a single byte value different from zero.

In the tweaked Grøstl-384/512, we increase the round constants to the following values:

$$P_{1024}: \ C[i] = \begin{bmatrix} \mathsf{00} \oplus i & \mathsf{10} \oplus i & \mathsf{20} \oplus i & \mathsf{30} \oplus i & \mathsf{40} \oplus i & \mathsf{50} \oplus i & \mathsf{60} \oplus i & \mathsf{70} \oplus i & \mathsf{80} \oplus i & \cdots & \mathsf{f0} \oplus i \\ \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \cdots & \mathsf{00} \\ \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \cdots & \mathsf{00} \\ \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \cdots & \mathsf{00} \\ \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \cdots & \mathsf{00} \\ \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \cdots & \mathsf{00} \\ \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \cdots & \mathsf{00} \\ \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \mathsf{00} & \cdots & \mathsf{00} \end{bmatrix}$$

and

$$Q_{1024}: \ C[i] = \begin{bmatrix} \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \cdots & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \cdots & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \cdots & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \cdots & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \cdots & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \cdots & \mathsf{ff} \\ \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \mathsf{ff} & \cdots & \mathsf{ff} \\ \mathsf{ff} \oplus i & \mathsf{ef} \oplus i & \mathsf{df} \oplus i & \mathsf{cf} \oplus i & \mathsf{bf} \oplus i & \mathsf{af} \oplus i & \mathsf{9f} \oplus i & \mathsf{8f} \oplus i & \mathsf{7f} \oplus i & \cdots & \mathsf{0f} \oplus i \end{bmatrix}$$

where i is the round number viewed as an 8-bit value, and all other values are written in hexadecimal notation.

3 Rationale for the Choice of Tweaks

Grøstl has received a large amount of cryptanalytic attention since it was submitted to the SHA-3 competition. Although there is no full cryptanalysis of the Grøstl hash functions, a possible conclusion from the published results is that the security of Grøstl would benefit from the permutations P and Q being more different. Therefore, we decided to make the described simple tweaks.

The change of shift values in Q means that differences propagate differently in P and Q. Some cryptanalysis of the previous version of Grøstl [2, 4–6, 8] has exploited the fact that by introducing a (truncated) difference in the input m, there is a relatively good probability that these differences propagate in the same way in P and Q, meaning that the output differences of P and Q may cancel each other. We have changed the shift values in Q such that no row is shifted by the same amount as in P, and such that the resulting states in P and Q are not simply shifted versions of each other. This way, it becomes much more difficult to ensure that output differences in P and Q may cancel each other.

Another type of cryptanalysis traces differences between P and Q, instead of the more traditional tracing of differences between pairs of inputs to a function [7]. This method has in particular been used in a collision attack on a reduced-round version of the Grøstl-0 hash function [3], and in the distinguishing attacks on the compression function [7] of Grøstl-0. By making the round constants different in all byte positions of the states of P and Q, these attacks are thwarted.

References

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

[2] H. Gilbert and T. Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In S. Hong and T. Iwata, editors, *Fast Software Encryption 2010, Proceedings*, volume 6147 of *Lecture Notes in Computer Science*, pages 365–383. Springer, 2010.

[3] K. Ideguchi, E. Tischhauser, and B. Preneel. Improved collision attacks on the reduced-round Grøstl hash function. In M. Burmester, G. Tsudik, S. Magliveras, and I. Ilic, editors, *Information Security Conference (ISC) 2010, Proceedings*, volume 6531 of *Lecture Notes in Computer Science*. Springer, 2011. To appear.

[4] 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 Jr., V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography 2009, Proceedings*, volume 5867 of *Lecture Notes in Computer Science*, pages 16–35. Springer, 2009.

[5] 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.

[6] F. Mendel, C. Rechberger, M. Schläffer, and S. S. Thomsen. Rebound Attacks on the Reduced Grøstl Hash Function. In J. Pieprzyk, editor, *Topics in Cryptology – CT-RSA 2010, Proceedings*, volume 5985 of *Lecture Notes in Computer Science*, pages 350–365. Springer, 2010.

[7] T. Peyrin. Improved Differential Attacks for ECHO and Grøstl. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010, Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 370–392. Springer, 2010.

[8] Y. Sasaki, Y. Li, L. Wang, K. Sakiyama, and K. Ohta. Non-full-active Super-Sbox Analysis: Applications to ECHO and Grøstl. In M. Abe, editor, *Advances in Cryptology – ASIACRYPT 2010, Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 38–55. Springer, 2010.