On the Complexity of Khovratovich et.al′s Preimage Attack on EDON-$\mathcal{R}$

Danilo Gligoroski and Rune Steinsmo Ødegård

March 12, 2009

Abstract

Based on the analysis made by van Oorschot and Wiener for the complexity of parallel memoryless collision search [5], we show that the memoryless meet-in-the-middle attack which is one part of the whole preimage attack of Khovratovich et. al. [3] on EDON-$\mathcal{R}$ hash function has complexity bigger than 2^n.

1 Introduction

For a proper understanding of the comments in this note the reader should be familiar with the notation that was used by Khovratovich, Nikolić and Weinmann [3]. Familiarity with the EDON-$\mathcal{R}$ hash function [2] is also strongly recommended.

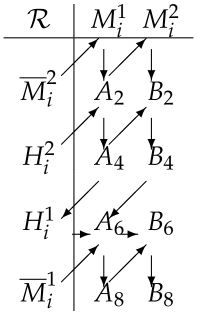

Table 1: Schematic representation of the compression function $\mathcal{R}$ as it is defined in EDON-$\mathcal{R}$.

In its compression function $\mathcal{R} : Q_q^4 \to Q_q^2$, EDON-$\mathcal{R}$ uses quasigroup operations $*_q$, $q = 256, 512$, (or shortly denoted as $*$) over the sets $Q_q = \{0,1\}^q$. The compression function is defined as:

$$\mathcal{R}(H_i^1, H_i^2, M_i^1, M_i^2) = (A_8, B_8) = (H_{i+1}^1, H_{i+1}^2)$$

where

$$\begin{aligned} A_8 &= \overline{M}_i^1 * ((H_i^2 * (\overline{M}_i^2 * M_i^1)) * H_i^1), \\ B_8 &= (\overline{M}_i^1 * ((H_i^2 * (\overline{M}_i^2 * M_i^1)) * H_i^1)) * (((H_i^2 * (\overline{M}_i^2 * M_i^1)) * \\ &\quad *((\overline{M}_i^2 * M_i^1) * M_i^2)) * ((H_i^2 * (\overline{M}_i^2 * M_i^1)) * H_i^1)). \end{aligned}$$

A graphical presentation of the compression function $\mathcal{R}$ is given in Table 1. The diagonal arrows can be interpreted as quasigroup operations between the source and the destination, and the vertical or the horizontal arrows as equality signs ″=″.

2 Description of attack

In what follows we will use the notation that was used in the FSE 2009 paper describing Khovratovich and Nikolić attack [3]. That means that the final output of the compression function $\mathcal{R}$ is denoted by the pair (H_2^1, H_2^2) where H_2^1 and H_2^2 are 256-bit or 512-bit values. The input message is denoted by the two pairs (M_0^1, M_0^2) and (M_1^1, M_1^2) where M_i^1 and M_i^2 are 256-bit or 512-bit values. In order to find a preimage for some predetermined value H_2^2 they launch a meet-in-middle (MITM) attack on two blocks of EDON-$\mathcal{R}$ where the forward direction is very fast, while the backward direction requires another MITM attack and is therefore very slow. We will first give a description of the forward direction.

$\mathcal{R}$	M_0^1	M_0^2
$\overline{M_0^2}$		
H_0^2		
H_0^1		
$\overline{M_0^1}$	0	H_2^{new}

a. 2^{n-s} different values of M_0^1

$\mathcal{R}$	m	
		$\boxed{B_3}$
	$\boxed{3}$	$\boxed{4}$
0	$\boxed{1}$	$\boxed{2}$
$\overline{m}$	A_8	H_2^2

b. $2^{\frac{n}{2}}$ different values of A_8

$\mathcal{R}$	m	M_1^2
$\overline{M_1^2}$	$\boxed{1}$	$\boxed{B_2}$
$\overline{m}$		

c. $2^{\frac{n}{2}}$ different values of M_1^2

Table 2: Schematic representation of the preimage attack of Khovratovich et. al. **a.** Forward direction of the main MITM attack **b.** Backward direction of the MITM attack on the second block. **c.** Forward direction of the MITM on the second block.

Forward direction. Using the fixed initial values H_0^1, H_0^2 of Table 2 a) and setting M_0^1 to some random value they compute M_0^2 such that $\mathcal{R}(M_0^1, M_0^2, H_0^1, H_0^2) = (H_1^1, H_1^2) = (0, H_2^{new})$ where H_2^{new} is some random value depending on M_0^1. This computation is fairly easy since the quasigroup operation used in the compression function of EDON-$\mathcal{R}$ is invertible. The attack requires 2^{n-s} such computations, where s is the number of times the backward direction is executed.

The backward direction is a MITM attack on the second block. Using a technique based on Floyd cycle finding first described in an article by Morita, Ohta and Miyaguchi [4] the memory requirements for this attack can be completely eliminated. This technique was later improved for parallel search by Oorschot and Wiener [5]. We will now describe the MITM on the second block, which is also the backward direction for the whole MITM attack.

Backward direction. The backward step in the MITM attack on the second block is performed by fixing B_8 to the desired hash H_2^2, fixing M_1^1 to some m and H_1^1 to 0 as shown in Table 2 b). Then different values of B_3 are computed by randomly setting $H_1^1 = A_8$ to some value and computing the intermediate values in the order shown in the table (boxes with numbers 1, 2, 3, 4 and B_3). In the terminology of van Oorschot and Wiener paper [5] we can represent this part of the attack as a mapping $f_2(A_8) = B_3$, where $f_2 : D_2 \to R$ and $|D_2| = |R| = 2^n$.

The forward step in the MITM attack on the second block is performed by setting M_1^2 to some random value and then computing B_2 using $M_1^1 = m$ in the order shown in Table 2 c). In the terminology of van Oorschot and Wiener paper this part of the attack is a mapping $f_1(M_1^2) = B_2$, where $f_1 : D_1 \to R$ and $|D_1| = 2^{n-65}$ (because of padding). Note that $|D_1| \leq |D_2|$ as required by Oorschot and Wieners's analysis of the parallel memoryless version of MITM attack.

In the version with use of huge memory, both f_1 and f_2 are evaluated $2^{\frac{n}{2}}$ times, and then intersection of two sets with $2^{\frac{n}{2}}$ elements each is performed in order to find a collision $B_2 = B_3$. Once a collision is found, it is used to calculate the corresponding $H_1^2 = H_2^{new'}$. This whole step is performed 2^s times to ensure that a collision $H_2^{new} = H_2^{new'}$ is found.

Khovratovich et. al., claim that using memoryless version of the meet-in-the-middle attack, this step requires $2^{n/2+s}$ calculations and no memory. As we will show, that is not true.

3 The cost of the memoryless MITM attack on the second block

In this analysis we will look closely at the complexity of the memoryless MITM attack on the second block. In [5] van Oorschot and Wiener gives a formula for the expected runtime of a parallel collision search algorithm that solves exactly the same problem that Khovratovich et. al., describe in their preimage attack on EDON-$\mathcal{R}$. In Section 5.3 of [5] van Oorschot and Wiener address the following problem: Given two functions $f_1 : D_1 \rightarrow R$ and $f_2 : D_2 \rightarrow R$, the goal is to find $a \in D_1$ and $b \in D_2$ such that $f_1(a) = f_2(b)$. The memoryless parallel algorithm that they describe has expected runtime, T_m:

$$T_m = (\frac{7n_2\sqrt{n_1/w}}{m})t \qquad (1)$$

where $|D_1| = n_1$, $|D_2| = n_2$, w is the buffer memory (shared by all processors) for which we assume that every processor has negligible $O(1)$ access time, m is the number of processors running in parallel and t is the time needed for one function iteration.

Setting $m = 1$, $n_1 = 2^{n-65}$, $n_2 = 2^n$ and $t = 1$, we get that the memoryless version of this MITM attack has the following expected number of computations of f_1 and f_2 functions:

$$2^{n+\frac{n}{2}-\frac{\log w}{2}-32.5+\log 7} \qquad (2)$$

To get a picture how much worse the preimage attack of Khovratovich et. al., compared by generic brute force attack is, let us put some concrete figures for the amount of the buffer memory used in their attack. Let us put the amount of buffer memory w to be unrealistically big like 2^{40} blocks i.e. 64 TBytes. Then, the expected number of computations will be $2^{n+\frac{n}{2}-69.69}$. For $n = 256$ we get that the number of computations will be bigger than $2^{314.31} \gg 2^{256}$ and for $n = 512$ the number of computations will be bigger than $2^{698.31} \gg 2^{512}$. Note that this is just one part of the whole preimage attack. This step is repeated 2^s times, obtaining 2^s collisions in the second block. From (2) we get that the memoryless version of the MITM attack on the second block has the following complexity:

$$2^{n+\frac{n}{2}-\frac{\log w}{2}-32.5+\log 7+s} \qquad (3)$$

The estimated complexity above is significantly bigger compared with the estimation $2^{n/2+s}$ that Khovratovich et. al. give in [3].

Finally, to estimate the complexity of the whole attack, we also need to take into account the 2^{n-s} evaluations of the compression function on the first block.

4 Conclusion

The complexity of the preimage attack of Khovratovich et. al. on EDON-$\mathcal{R}$ is not even close to the complexity of the generic preimage attack on any n-bit hash function.

From this perspective, the wording that there is no similar attack on SHA-2 (found for example on SHA-3 Zoo web pages[1]) is completely misleading and irrational. Maybe it is true that there is no similar attack on SHA-2 as Khovratovich et. al. attack on EDON-$\mathcal{R}$, but there is much faster attack on SHA-2. The name of that attack is: Brute force attack.

Acknowledgement: The first author of this note wants to thank Ivica Nikolić for long and fruitful discussions about Khovratovich and Nikolić preimage attack on EDON-$\mathcal{R}$, during FSE 2009 and The First SHA-3 Conference, held in Leuven in period from 23 Feb. 2009 – 28 Feb. 2009.

References

[1] `http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`.

[2] Danilo Gligoroski, Rune Steinsmo Ødegård, Marija Mihova, Svein Johan Knapskog, Ljupco Kocarev, Aleš Drápal, and Vlastimil Klima. Cryptographic hash function EDON-$\mathcal{R}$. Submission to NIST, 2009.

[3] Dmitry Khovratovich, Ivica Nikolić, and Ralph-Philipp Weinmann. Meet-in-the-middle-attack on SHA-3 canditates. In *Fast Software Encryption - 2009*, pages 233–250, 2009.

[4] Hikaru Morita, Kazuo Ohta, and Shoji Miyaguchi. A switching closure test to analyze cryptosystems. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 183–193, London, UK, 1992. Springer-Verlag.

[5] Paul C. Van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12:1–28, 1999.