Practical Collision and Preimage Attack on DCH-n

Mario Lamberger and Florian Mendel

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology, Inffeldgasse 16a, A-8010 Graz, Austria

Abstract. In this paper, we show practical collision and preimage attacks on DCH-n. The attacks are based on the observation of Khovratovich and Nikolic that the chaining value is not used in the underlying block cipher. Based on this observation, we show a trivial collision resp. multi-collision attack on DCH-n and a preimage attack with a complexity of about 583 compression function evaluations.

1 Description of DCH-n

The hash function DCH-n is an iterated hash function based on the Merkle-Damgaard design principle. It processes message blocks of 512 bits (504 bits message input) and produces a hash value of $n = 224, 256, 384$ or 512 bits. In each iteration the compression function f is used to update the chaining value of 512 bits as follows:

$$H_{i+1} = f(H_i, M_i) = H_i \oplus M_i \oplus g(M_i) \ ,$$

where $g(M)$ is some non-linear transformation. For a detailed description of DCH-n we refer to [3].

2 Cryptanalysis

In this section, we will present our collision and preimage attack on DCH. The attack is an extension of the attack of Khovratovich and Nikolic [1] and is based on similar principles as the attacks on SMASH [2]. Let $\gamma_i(M_i) = g(M_i) \oplus M_i$. Then the above equation can be rewritten as:

$$H_i = H_0 \oplus \gamma_0(M_0) \oplus \gamma_1(M_1) \oplus \cdots \oplus \gamma_i(M_i)$$

Note that the γ_i are different since in DCH-n a block counter is used in each message block to compute $M_i \oplus g(M_i)$. However, this counter is reset to 0 after computing 32 message blocks. Hence, we know that $\gamma_i = \gamma_j$ for $i \equiv j \pmod{32}$. Based on this observation, we now introduce an alternative description of DCH-n. Let $\Gamma(m_0) = \gamma_0(M_0) \oplus \gamma_1(M_1) \oplus \cdots \oplus \gamma_{31}(M_{31})$ then $H_{32} = H_0 \oplus \Gamma(m_0)$ with $m_0 = M_0 \| M_1 \| \cdots \| M_{31}$. In general, we have

$$H_{(i+1)\cdot 32} = H_0 \oplus \Gamma(m_0) \oplus \cdots \oplus \Gamma(m_i) \ ,$$

with $m_i = M_{32\cdot i} \| M_{32\cdot i+1} \| \cdots \| M_{32\cdot i+31}$.

2.1 Collision Attack

Based on this alternative description of DCH-n, we now describe the collision attack. Assume we have given a message $M = m_0 \| m_1$ consisting of $(32 \cdot 63) \cdot 2$ bytes. Then the chaining value $H_{64} = H_0 \oplus \Gamma(m_0) \oplus \Gamma(m_1)$. Furthermore, let $m_1 = m_0$ then $H_{64} = H_0$. Hence, constructing a collision in DCH-n is easy.

1. Choose an arbitrary value for m_0 and compute H_{64} with $m_1 = m_0$.
2. Choose an arbitrary value for $m_0^* \neq m_0$ and compute H_{64} with $m_1^* = m_0^*$. It is easy to see that this leads to a collision for $m_0 \| m_1$ and $m_0^* \| m_1^*$ with $H_{64} = H_{64}^* = H_0$.

Hence, we can trivially construct collisions for DCH-n. Note that the messages in the colliding message pair consist of 2^6 message blocks. Furthermore, we can trivially construct t-collisions (for $0 < t < 2^{32 \cdot 63}$) for DCH-n, since there exists many possible choices for m_0 in our attack. Note that all these attacks apply to DCH-n for all output sizes.

2.2 Preimage Attack

In a similar way as in the collision attack, we can also construct preimages for DCH-n. The attack is based on the observation that the outputs of DCH-n form a vector space of dimension n over $GF(2)$ (cf. also [2]). Hence, we only need to compute a basis of the output vector space to construct preimages for DCH-n. In the following we set $N := 512 \cdot 32 \cdot 2 = 2^{15}$. Furthermore, we assume $n = 512$ since the other output lengths result from truncations of the $n = 512$ version. Then, the attack can be summarized as follows:

1. Assume we want to construct a preimage for h consisting of $N + 1$ message blocks. Then, we have to find a message M such that:

$$h = H_0 \oplus \bigoplus_{i=0}^{N} \gamma_{i \bmod 32}(M_i) \ .$$

2. Choose the last message block M_N such that the padding is correct.
3. Once, we have fixed the last message block, we have to find the remaining message blocks M_i for $0 \leq i < N$ such that:

$$\bigoplus_{i=0}^{N-1} \gamma_{i \bmod 32}(M_i) = h \oplus H_0 \oplus \gamma_0(M_N) \ .$$

   For simplicity, let us now use the alternative description of DCH-n. Then the above equation can be written as:

$$\bigoplus_{i=0}^{N/32-1} \Gamma(m_i) = c \ ,$$

   where $c = h \oplus H_0 \oplus \gamma_0(M_N)$ and $m_i = M_{32 \cdot i} \| M_{32 \cdot i+1} \| \cdots \| M_{32 \cdot i+31}$. To solve this equation, we use now the fact that the outputs of DCH-n form a vector space.

4. Compute ℓ vectors $a^k = \Gamma(m_0^k) \oplus \Gamma(m_1^k)$ with arbitrary values for m_0 and m_1 and save the triple (a^k, m_0^k, m_1^k) in a list L.
5. From the set of $\ell \geq n$ vectors a^k compute a basis of the output vector space of DCH-n. The probability for $\ell \geq n$ vectors to contain n vectors which are linearly independent is

$$\prod_{i=0}^{n-1} \frac{2^\ell - 2^i}{2^\ell} = \prod_{i=0}^{n-1} (1 - 2^{i-\ell}) \geq 2^{-\frac{2^n-1}{2^{\ell-1}}} .$$

   This means that we can basically construct such a basis with complexity of $64 \cdot \ell$ compression function evaluations. This can be reduced to $63 + \ell$ evaluations of the compression function by fixing all blocks in m_0^k and all but one block in m_1^k when generating the basis of the output vector space. For example choosing $n = 512$ and $\ell = 520$ we already get a probability of 0.9961 for finding a basis and thus need only 583 compression function evaluations. Note, that constructing the basis is a one time effort. Let $B = \{a^{k_0}, \ldots, a^{k_{n-1}}\}$ denote the basis for the output vector space.
6. We then represent c with respect to this basis $c = x_0 a^{k_0} + \cdots + x_{n-1} a^{k_{n-1}}$ by solving the linear system over $GF(2)$.
7. Next, we use the x_j to construct $m_0, m_1, \ldots, m_{1023}$ such that:

$$\bigoplus_{i=0}^{1023} \Gamma(m_i) = c \ .$$

   – If $x_j = 0$ for $0 \leq j < 512$ set $m_{2j} = \alpha$ and $m_{2j+1} = \alpha$ for some arbitrary value of α. Note that $\Gamma(\alpha) \oplus \Gamma(\alpha) = 0$ and hence, m_{2j} and m_{2j+1} have no influence on the computation of c.
   – If $x_j = 1$ for $0 \leq j < 512$ set $m_{2j} = m_0^j$ and $m_{2j+1} = m_1^j$ such that $\Gamma(m_0^j) \oplus \Gamma(m_1^j) = a^j \ 0 \leq j < 512$.
   – This technicality is necessary because we want to construct a preimage of fixed length.

Hence, we can construct a preimage for DCH-n by solving a linear system of equations of dimension 512×512 over $GF(2)$. Constructing the basis has a complexity of about 583 compression function evaluations.

Furthermore, the preimage attack can be used to construct second preimages for DCH-n with the same complexity. Note that by using the above described method, preimages (or second preimages) always consist of $N + 1 = 2^{15} + 1$ message blocks.

3 Conclusion

We showed, that it is trivial to construct collisions and (second) preimages for DCH-n. Furthermore, the presented attack applies to all similar constructions not introducing the chaining variable into the compression function.

References

1. Dmitry Khovratovich and Ivica Nikolic. Cryptanalysis of DCH-n, 2008. Available online: `http://lj.streamclub.ru/papers/hash/dch.pdf`.
2. Mario Lamberger, Norbert Pramstaller, Christian Rechberger, and Vincent Rijmen. Analysis of the hash function design strategy called smash. *IEEE Transactions on Information Theory*, 54(8):3647–3655, 2008.
3. David A. Wilson. The DCH Hash Function. Submission to NIST, 2008. Available online: `http://web.mit.edu/dwilson/www/hash/dch/Supporting_Documentation/dch.pdf`.