An Observation on JH-512

Florian Mendel[1] and Søren S. Thomsen[2]

[1] Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Inffeldgasse 16a, A-8010 Graz, Austria.
florian.mendel@iaik.tugraz.at
[2] Department of Mathematics, Technical University of Denmark
Matematiktorvet 303S, DK-2800 Kgs. Lyngby, Denmark.
crypto@znoren.dk

Abstract. In this paper, we present a generic preimage attack on JH-512. We do not claim that our attack breaks JH-512 (due to the high memory requirements), but it uses some interesting properties in the design principles of JH-512 which do not exist in other hash functions, *e.g.*, the SHA-2 family.

1 Description of JH

The hash function JH is an iterated hash function. It processes message blocks of 512 bits and produces a hash value of 224, 256, 384, or 512 bits. In each iteration the compression function f is used to update the chaining value of 1024 bits as follows:

$$H_i = f(H_{i-1}, M_i)$$

where H_{i-1} is the previous chaining value, M_i is the current message block. The compression function f is defined as follows:

$$f(H_{i-1}, M_i) = E(H_{i-1} \oplus M_i \| 0^{512}) \oplus 0^{512} \| M_i$$

where E is a permutation of 1024 bits, and 0^{512} means the string of 512 '0' bits. The details of E are irrelevant to the attack described in this paper, but we assume that the outputs of f are roughly Poisson distributed when H_{i-1} is fixed.

After the last message block has been processed, the final hash value is generated from the last chaining value by truncation. For a detailed description of JH we refer to [**?**].

2 Generic Preimage Attack

In this section, we present a preimage attack on JH-512 with complexity of about $2^{510.3}$ compression function evaluations and the same amount of memory. The attack is based on the following two observations on the compression function f.

Observation 1. The compression function f is invertible, meaning that given H_i and M_i, it is easy to find H_{i-1} such that $f(H_{i-1}, M_i) = H_i$, namely as $H_{i-1} = E^{-1}(H_i \oplus 0^{512} \| M_i) \oplus M_i \| 0^{512}$.

Hence, pseudo-collisions and pseudo-preimages can be found trivially [**?**].

Observation 2. For arbitrary H_{i-1}, M_i and $H^*_{i-1} = H_{i-1} \oplus \Delta \| 0^{512}$, $M^*_i = M_i \oplus \Delta$, the following relation holds:

$$f(H_{i-1}, M_i) \oplus f(H^*_{i-1}, M^*_i) = 0^{512} \| \Delta$$

for any choice of Δ.

Furthermore, the attack makes use of *multicollisions*.

Definition 1. *Let g be some function. An r-collision for g is an r-set $\{x_1, \ldots, x_r\}$ such that $g(x_1) = \ldots = g(x_r)$. A multicollision is an r-collision for some $r > 1$.*

If g is a random n-bit function, then finding an r-collision in g has a complexity of about

$$q = (r! \cdot 2^{n(r-1)})^{1/r} \tag{1}$$

evaluations of g [**?**]. This estimate can be obtained from the Poisson formula $F(r, \lambda) = \lambda^r \exp(-\lambda)/r!$ by using $\lambda = q2^{-n}$ and setting $F(r, \lambda) = 2^{-n}$. Furthermore, the factor $\exp(-\lambda)$ is removed, since it is very close to 1 when $q \ll 2^n$. Finding ℓ r-collisions requires only a factor about $\ell^{1/r}$ more work than finding a single r-collision, which is seen by setting $F(r, \lambda) = \ell 2^{-n}$.

We will use this to construct preimages for JH-512 with a complexity of about $2^{510.3}$. Assume we want to construct a preimage for the 512-bit target image h. The preimage will consist of 4 message blocks. The attack can be summarised as follows.

1. Choose an arbitrary message block M_4 with correct padding, and compute $H_3 = f^{-1}(x \| h, M_4)$ for an arbitrary 512-bit value x.
2. Compute 2^{509} candidates for $H_2 = f^{-1}(H_3, M_3)$ with arbitrary choices of M_3, and save the pairs (H_2, M_3) in a list L.
3. Use M_1 to construct an r-collision for the 512 higher bits of H_1, given the initial value H_0 of JH-512. For $r = 51$ this has a complexity of about $2^{506.3}$ compression function evaluations. In other words, we find $r = 51$ message blocks M_1^k for $0 \leq k < r$ such that b^k is equal with $H_1^k = a^k \| b^k$.
4. Compute $\Delta^k = H_1^0 \oplus H_1^k$ for $0 \leq k < r$.
5. Choose an arbitrary message block M_2 and compute $H_2 = f(H_1^0, M_2)$ and check if $H_2^k = H_2 \oplus \Delta^k$ for $0 \leq k < r$ is in the list L. The probability for each choice of M_2 is about $51 \cdot 2^{1024-509}$, so we need to try an expected $2^{515}/51 \approx 2^{509.3}$ message blocks. Note that only about $2^{512}/51 \approx 2^{506.3}$ different message blocks can be chosen in this step without repetition, and hence we must find an expected 2^3 51-collisions in step 3. However, 2^3 51-collisions can be found in time only a factor about $2^{3/51} \approx 2^{0.06}$ more than a single 51-collision. Thus, the "new" complexity of step 3 is $2^{506.3}$ (unchanged to one decimal place), and the current step has complexity about $2^{509.3}$ (we ignore the 51 xors needed in this step, assuming this takes negligible time compared to one evaluation of f).
6. Once we have found H_2^k such that a pair (H_2^k, M_3) is in the list L, we have to adjust M_1 and M_2 accordingly such that $f(f(H_0, M_1), M_2) = H_2^k = H_2 \oplus \Delta^k$.
   It is easy to see that this can be achieved by setting $M_1 = M_1^k$ and $M_2 = M_2 \oplus \Delta^k$, since:

$$\begin{aligned} H_1 &= f(H_0, M_1^k) = H_1^k = H_1^0 \oplus \Delta^k \\ H_2 &= f(H_1^0 \oplus \Delta^k, M_2 \oplus \Delta^k) = H_2 \oplus \Delta^k = H_2^k \end{aligned}$$

Hence, we can find a preimage for JH-512 with a total complexity of about $2^{509} + 2^{506.3} + 2^{509.3} \approx 2^{510.3}$ compression function evaluations and a similar amount of memory. Note that the attack complexity might be higher than brute force search in practice due to the high memory requirements and the number of needed memory accesses. Nevertheless, we think that the attack shows some interesting properties of JH-512, which do not exist in other hash functions. Maybe these properties can be combined with a dedicated preimage attack on JH-512 in the future. At the moment, our attack does not compromise the security claims of JH-512.