Un*tangle*d

Søren S. Thomsen

crypto@znoren.dk

DTU Mathematics, Technical University of Denmark

December 16, 2008

Abstract

We describe a collision attack on all variants of the SHA-3 candidate Tangle. Practical collisions have been found, and memory requirements are negligible. The time complexity of the attack ranges from a few seconds (equivalent to 2^{13}–2^{19} compression function evaluations) for all except the longest variant, to several minutes, or about 2^{28} compression function evaluations for the longest (1024-bit) variant.

1 Introduction

Tangle [1] is a candidate for NISTs SHA-3 competition [2]. In this paper, we describe a collision attack on Tangle. The attack applies to all digest sizes. The time complexity is very low in most cases (finding a collision takes only seconds for all digest sizes up to 768 bits), and memory requirements are negligible.

For a detailed description of Tangle, we refer to [1].

2 The collision attack

Tangle is capable of returning digests of sizes 224, 256, 384, 512, 768, and 1024 bits. All variants of Tangle operate fundamentally in the same way, only initial values, the number of rounds, and the amount of truncation done in the end are different. The message block size is 4096 bits, separated into 128 32-bit words. If the number of rounds is R, then the 4096-bit message block is expanded into $2R$ 32-bit words. The details of the message expansion shall not be described here, but we mention that flipping the most significant bit (MSB) of a number of message block words in a certain pattern has the effect that a number of words in the *expanded* message have their MSBs flipped, but all other bits are unaffected.

The state consists of 32 words of 32 bits each, in total 1024 bits. The number of rounds depends on the digest size, but is between 72 (for the 224-bit digest size) and 144 (for the 1024-bit digest size). Let the expanded message words be denoted W_i, $0 \leq i < 2R$, and let the 32 state words be denoted h_i, $0 \leq i < 32$. Let K_i, $0 \leq i < 256$, be 32-bit constants. Let S be an 8-bit s-box, whose details are irrelevant to this attack. The round function performs the following (addition is to be taken modulo 2^{32}, $x^{\gg \ell}$ means x shifted right by ℓ bit positions, i is the round number, and s is initially zero):

$$\begin{array}{rcl} C & \leftarrow & W_{2i} + W_{2i+1} \\ s & \leftarrow & s \oplus S(C \oplus C^{\gg 8} \oplus C^{\gg 16} \oplus C^{\gg 24}) \\ p & \leftarrow & s \bmod 16 \\ q & \leftarrow & s^{\gg 4} \bmod 16 \\ A & \leftarrow & F_{i \bmod 2+1}(h_p, h_{i \bmod 32}, FR_1(h_{q+16})) + W_{2i} + K_s \\ B & \leftarrow & F_{i \bmod 2+1}(h_q, h_{i+8 \bmod 32}, FR_2(h_{p+16})) + W_{2i+1} \\ h_{i \bmod 32} & \leftarrow & h_{i \bmod 32} + B \\ h_{i+16 \bmod 32} & \leftarrow & h_{i+16 \bmod 32} \oplus (A + B) \end{array}$$

The functions F_1 and F_2 are balanced, bitwise logical functions. Hence, a difference in a certain bit position in any of their inputs will stay in that bit position, or be cancelled out. They are defined as follows:

$$\begin{array}{rcl} F_1(x, y, z) & = & (x \wedge (y \vee z)) \vee (y \wedge z) \\ F_2(x, y, z) & = & (\neg y \wedge (x \vee z)) \vee (x \wedge z). \end{array}$$

The functions FR_1 and FR_2 are xors of three differently rotated versions of their input. The details of FR_1 and FR_2 are irrelevant to this attack, since there will never be a difference in the inputs to FR_1 or FR_2 in colliding messages. However, this will be ensured somewhat probabilistically.

Assume that two different messages contain a difference in the most significant bits of both W_{2i} and W_{2i+1}. Then these differences cancel out in the first computation in the round function, and there is no difference on s, p or q. Moreover, assuming no difference on any of the state words, there will be a difference in the MSB of A and B. These differences will cancel out in the expression $A + B$ which appears in the last computation. Hence, a difference will be added to the MSB of $h_{i \bmod 32}$ only. It is easy to ensure that there is a difference in the MSB of W_{2i} if and only if there is a difference in the MSB of W_{2i+1}.

A difference in the MSB of a state word not entering either of FR_1 or FR_2 may survive the function $F_{i \bmod 2+1}$, or it may cancel out.

These effects on the most significant bit can be used to form collisions. One may choose a pattern of differences in the MSB of message words such that the differences in the expanded message words are as desired. Given this pattern, one may search for absolute values of message words until differences in the MSBs cancel out in desired state words. Although we have found messages that collide in the entire state, also an internal near-collision may extend to a full collision after truncation. Therefore, the collision search has a very low complexity for all digest sizes up to 768 bits.

As an example, consider Tangle-256, for which the number of rounds, and therefore half the number of expanded message words, is 80. We find collisions in Tangle-256 between 40-byte messages (note that a 40-byte message is padded to one message block). We may choose to add differences in the MSBs of message words no. 0, 1, 8, and 9, resulting in differences in the MSBs of expanded message words 0, 1, 8, 9, 128, 129, 136, and 137. We let message words 1–9 of the first message be equal to 0, and search for increasing values of message word 0 until a collision between two messages with the given difference pattern is found. This occurs when message word 0 of the first message has the integer value 6600, or `19c8` in hexadecimal.

Collisions in all Tangle variants can be found in the same way, using the same pattern of message differences. Example collisions are given in Table 1. An estimate on the search complexity is also given by the table, since increasing values of word no. 0, starting from 1, were tried. E.g., for Tangle-384, about $2 \cdot \texttt{51376}_h \approx 2^{19}$ evaluations of the compression function were needed, whereas for Tangle-768, only about 2^{13} evaluations were needed. For Tangle-1024, where the collision must extend the entire state, the complexity was about 2^{28}. Patterns yielding a lower complexity are likely to exist.

Table 1: Collisions for different Tangle variants (values of message word 0 are in hexadecimal).

Digest size	Message word 0	Message words 1–9	Diff. in MSB of words
224	`3e83f`	0	0, 1, 8, 9
256	`19c8`	0	0, 1, 8, 9
384	`51376`	0	0, 1, 8, 9
512	`a9ab`	0	0, 1, 8, 9
768	`16d3`	0	0, 1, 8, 9
1024	`721bdfb`	0	0, 1, 8, 9

3 Conclusion

Since Tangle allows practical collisions to be found, apparently independently of the number of rounds, we believe that Tangle does not fulfil the requirements for a SHA-3 candidate.

The collision search program can be downloaded from `http://www.mat.dtu.dk/people/S.Thomsen/tangle/tangle-coll.c` (must be compiled together with the Tangle reference implementation from the submission package).

References

[1] R. Alvarez, G. McGuire, and A. Zamora. The Tangle Hash Function. SHA-3 submission, 2008. Available: `http://ehash.iaik.tugraz.at/uploads/4/40/Tangle.pdf` (2008/12/15).

[2] National Institute of Standards and Technology. Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family. *Federal Register*, 27(212):62212–62220, November 2007. Available: `http://csrc.nist.gov/groups/ST/hash/documents/FR_Notice_Nov07.pdf` (2008/10/17).