Differential Cryptanalysis of KHF

David Wagner

University of California, Berkeley
daw@cs.berkeley.edu

Abstract. Bakhtiari et al recently proposed a fast message authentication primitive called KHF. This paper shows that KHF is highly vulnerable to differential cryptanalysis: it can be broken with about 37 chosen message queries. This suggests that the KHF design should be reconsidered.

1 Introduction

Recent applications in secure networking have spurred research in cryptographic primitives for message authentication. In particular, there is great demand for a high-speed MAC that can be implemented in software. In 1996, Bakhtiari, Safavi-Naini, and Pieprzyk proposed a new MAC primitive, called KHF, for fast software message authentication [BSF95]. See Section 2 for a brief description of their primitive.

One of the main contributions of the KHF work is its careful attention to using Boolean functions which can be shown to have very good non-linearity, as well as other desirable theoretical properties. Nonetheless, this paper shows how to break KHF efficiently, despite its solid theoretical foundation in cryptographic Boolean function theory.

In this paper we show how to break KHF with differential cryptanalysis [BS93]. The attack requires just 37 chosen messages, and is described in Section 3.

2 Description of KHF

We give a brief overview of the KHF message authentication algorithm here, omitting those details that are irrelevant to our attack.

Before processing, padding is prepended and appended to the message, and the result is split into 32-bit blocks denoted by $M_1, M_2, \ldots, M_n$. Two 128-bit state buffers ($X_{1\ldots4}$ and $Y_{1\ldots4}$) and a 512-bit redundancy buffer ($B_{1\ldots16}$) are used internally. The key is fed into the initial values of the X, Y buffers and optionally also into the padding, with the exact details depending on the mode of operation.

We perform n rounds, one for each message block. The i-th round uses M_i, B_i, and Y to update the three buffers: it calculates $T = f_i(Y_1, Y_2, Y_3, Y_4, M_i)$, modifies X using T, xors M_i into B_{a_i} (where a_i is derived from T), and then

S. Vaudenay (Ed.): Fast Software Encryption – FSE'98, LNCS 1372, pp. 293–296, 1998.

modifies Y using T and B_i. (These modifications do not depend implicitly on any unmentioned variables.)

The round functions f_i are defined as

$$f_i(A,B,C,D,E) = \begin{cases} (A\&E) \oplus (B\&C) \oplus ((B \oplus C)\&D) & i = 1 \bmod 5 \\ A \oplus (B\&(A \oplus D)) \oplus (((A\&D) \oplus C)\&E) & i = 2 \bmod 5 \\ A \oplus (C\&D\&E) \oplus ((A\&C)|(B\&D) & i = 3 \bmod 5 \\ B \oplus ((D\&E)|(A\&C)) & i = 4 \bmod 5 \\ D \oplus E \oplus (((D\&E) \oplus A)\& \sim (B\&C)) & i = 0 \bmod 5 \end{cases}$$

where the logical operations $\oplus, \&, |, \sim$ are performed on each bit of the 32-bit words independently. Here $\oplus$ represents bitwise exclusive-or, & represents bitwise and, | represents bitwise or, and $\sim$ represents bitwise logical negation.

3 Analysis

First, we exhibit a useful high-probability differential characteristic [BS93] for KHF.

The main observation is that a one-bit difference into the input of the round function f is very likely to yield a zero output difference (i.e. a collision). This arises from the fact that the f_i operate on each bit position independently; in other words, the KHF round functions have very poor diffusion properties across bit positions. More precisely, flipping one bit in E leaves $f_i(A,B,C,D,E)$ unchanged with the probabilities given below:

i mod 5	1	2	3	4	0
Prob	1/2	1/2	3/4	5/8	3/8

This corresponds to a differential characteristic for f_i of the form $(0,0,0,0,2^j) \mapsto 0$. Such characteristics hold with very high probabilities despite the focus on the construction of f from highly non-linear Boolean function theory.

Next, we show how to extend the high-probability differential characteristic for the f function into a characteristic for the whole algorithm. Forcing such a difference into one M_i leaves $T = f_i(Y_1, Y_2, Y_3, Y_4, M_i)$ unchanged, and thus leaves the X and Y buffers unchanged; however, the B_{a_i} word does change. Therefore, to obtain a collision for the entire KHF calculation, we introduce the same one-bit difference into two message words, M_i and M_j where $i < j$. While this introduces a difference into the B buffer at the i-th round, we hope that it gets canceled out later in the j-th round before the difference has a chance to propagate. This gives us two conditions for success: first, we must have $a_i = a_j$; second, we require that B_{a_i} is not present in the list $B_i, B_{i+1}, \ldots, B_{j-1}$. (Subscripts on B are taken modulo 16.)

To calculate the characteristic's probability, we can model the a_i as effectively random integers selected from the set $\{1, 2, \ldots, 16\}$, since we do not know the value of T or Y. For best chance of success we suggest taking $j = i + 1$ and

$i = 3 \bmod 5$; then the probability that both conditions hold is

$$\frac{3}{4} \cdot \frac{5}{8} \cdot \frac{15}{16} \cdot \frac{1}{16} = \frac{225}{8192} \approx \frac{1}{36}.$$

To recap, introducing the same one-bit difference into M_i and M_{i+1} (for some $i = 3 \bmod 5$) will yield a collision in the final output of KHF with this 1/36 probability. We have implemented the attack and empirically confirmed this analysis.

We now describe how to break KHF using differential cryptanalysis. We have given a class of differences which, when xored into the message input, yield a collision in the KHF output with very high probability. This makes breaking the KHF message authentication scheme easy. For example, to forge the MAC digest $\text{KHF}(K, M)$ on a message M under a chosen-plaintext threat model, we could obtain the MAC digests $D_i = \text{KHF}(K, M \oplus \Delta_i)$ (where each Δ_i is one of the xor differences of Hamming weight two described above) and look for repetitions in the list of D_i. With high probability the repeated value will be the desired MAC digest $\text{KHF}(K, M)$. By using the high-probability characteristic described above, this attack will need only about 72 chosen plaintext queries[1] to break a system using KHF for message authentication.

Other attacks are also possible. For example, to learn $\text{KHF}(K, M||Y)$ (where $||$ denotes concatenation on 32-bit boundaries), we offer the following differential attack needing 37 chosen plaintext queries. Find M' (of the same length as M) so that $\text{KHF}(K, M||X) = \text{KHF}(K, M'||X)$ (where X has the same length as Y but is otherwise arbitrary), by using the differential attack developed above. The collision $\text{KHF}(K, M||X) = \text{KHF}(K, M'||X)$ typically arises because the internal states after processing M and M' match. Now use a single chosen plaintext query to learn $\text{KHF}(K, M'||Y)$. The relation $\text{KHF}(K, M||Y) = \text{KHF}(K, M'||Y)$ will hold with very high probability; this lets us deduce $\text{KHF}(K, M||Y)$, as desired. Note that we have never used $M||Y$ in a chosen-message query, but we obtain the MAC digest corresponding to $M||Y$, so we have broken the MAC primitive. The attack lets us break the KHF message authentication scheme with about 37 chosen plaintexts, on average.

This latter attack is based on the observation that "internal collisions" can lead to MAC forgery. Of course, this property is hardly novel; it was one of the central techniques used to break a number of MAC primitives in [PO95].

4 Conclusion

KHF was designed around a set of Boolean functions with excellent non-linearity properties. The round function f on 32-bit input words was constructed by applying a good Boolean function to each bit position independently. This ensures

[1] One caveat: this assumes that the message is sufficiently long (60 bytes) so that we can easily find 72 two-bit Δ_i values of the right form. Shorter messages can also be attacked with similar differential characteristics that have a somewhat lower probability; the disadvantage is that we will need slightly more chosen plaintext queries.

that we achieve the best possible diffusion within any one bit position, but its fatal flaw is that it gives very poor diffusion across bit positions.

Our cryptanalysis of KHF takes advantage of this weakness in the round function. We note that this design process—start with a good Boolean function and extend it to a n-bit function by bitwise parallel evaluation—has fundamental flaws: no matter how many good properties the Boolean function has, the full n-bit round function will have very poor diffusion across bit positions. Paying too much attention to the theoretical properties of the underlying Boolean function can be dangerous, if that causes the rest of the algorithm structure to receive less attention. It is important to start from a solid foundation, but that alone is not enough.

To summarize, KHF suffers from serious vulnerabilities to differential attack, and should be considered insecure. The flaws that we have found do not inspire confidence in the design process, nor is it clear whether there is any simple way to fix these flaws without radically modifying the structure of KHF. Therefore, we recommend that the KHF design be abandoned.

References

BSF95. S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk, "Keyed hash functions," *Cryptography: Policy and Algorithms*, E. Dawson and Jovan Golic (Eds), Lecture Notes in Computer Science, vol. 1029, Springer-Verlag, 1996, pp.201-214.

BS93. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

PO95. B. Preneel, P.C. van Oorschot, "MDx-MAC and building fast MACs from hash functions," *Advances in Cryptology—CRYPTO '95*, Springer-Verlag, 1995, pp. 1–14.