Presented at Eurocrypt '92, Balatonfüred, Hungary.

Differential Cryptanalysis Mod 2^{32} with Applications to MD5.

Thomas A. Berson
Anagram Laboratories
P.O. Box 791
Palo Alto, CA 94301, USA

Abstract

We introduce the idea of differential cryptanalysis mod 2^{32} and apply it to the MD5 message digest algorithm. We derive a theory for differential cryptanalysis of the circular shift function. We demonstrate a high-probability differentials which leave the message digest register unchanged for each of MD5's four rounds, and explain how more such differentials may be calculated.

1 Introduction

Differential cryptanalysis is a method which analyses the effect of particular differences in plaintext pairs on the differences of the resulting ciphertext pairs. Since differential cryptanalysis was first explained by Biham and Shamir [BS1] [BS2] it has been applied with success, sometimes with devastating success, to cryptosystems including DES [BS1], [BS2], Feal [BS3], N-Hash [BS3], PES [LMM], Snefru [BS4] Khafre [BS4], REDOC-II [BS4], LOKI [BS4] [Knud] [BKPS], and Lucifer [BS4]. A common element in these cryptosystems which makes them susceptible to differential cryptanalysis is their heavy use of the exclusive-or operation (denoted by $\oplus$ or XOR), which is equivalent to vector addition mod 2, to introduce confusion in combining partial results or in combining key with data. Differential cryptanalysis is able to reduce or negate the cryptographic effects of these XORs by considering the differences in ciphertexts which arise from operating the cryptosystem on pairs of plaintexts chosen so that they are at a fixed distance (mod 2) from one another. The appropriate fixed distance changes from cryptosystem to cryptosystem, and is derived by analysis of the structure of the cryptosystem.

Designers of cryptosystems published since the rise of differential cryptanalysis have sought to avoid its sting by reducing or avoiding the use of XOR in its traditional roles. The message digest algorithm MD5 [Riv] [RD] is an example of such a post-differential cryptosystem. MD5 is designed to be fast on 32-bit machines, and employs addition mod 2^{32} (denoted by +) to achieve confusion.

Unfortunately for designers, there is nothing which binds differential cryptanalysis to any particular algebraic group. In this paper we apply differential cryptanalysis mod 2^{32} to MD5.

So far as possible, we will follow the notation of Biham and Shamir [BS2] and Rivest and Dusse [RD]. In particular:

Y^*, Y': At any point during the operation of the MD5 algorithm on pairs of messages, Y and Y^* are the values of the two executions of the algorithm, and Y' is defined to be $Y' = Y - Y^* \bmod 2^{32}$, $0 \leq Y' \leq 2^{32} - 1$.

$\tilde{M}$: the input message, b bits in length. b is any non-negative integer.

n_x: A hexidecimal number is denoted by the subscript x.

2 The MD5 Message Digest Algorithm

The MD5 message digest algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest. MD5 is intended by its authors for use in digital signature applications. These applications require that it be computationally infeasible to produce two messages having the same digest, for it is the digest which is signed, not the original message.

2.1 Overall Structure

The computation of the message digest of $\tilde{M}$ by MD5 is carried out in five stages.

Stage 1. The message $\tilde{M}$ is padded (extended) so that its length in bits is congruent to 448 modulo 512. The padding consists of a single "1" bit followed by as many "0" bits as necessary to reach the desired length.

Stage 2. A 64-bit representation of b, the length of $\tilde{M}$ before padding, is appended to the results of Stage 1. The resulting message M has a length which is an exact multiple of 512 bits. Equivalently, this message has a length which is an exact multiple N of 16 32-bit words.

Stage 3. A four-word register *MD* is used to compute the message digest. This is initialized to a constant.

Stage 4. The message M is processed in consecutive *blocks* of 16 words, M_1, M_2, ... , M_{N-1}, M_N. The processing of each block consists of four *rounds*, each of which consists of 16 *steps*. We will have much more to say about rounds and steps below.

Stage 5. Register *MD* now contains the calculated message digest. This is output.

In seeking two equivalent messages we will work within a single 16-word block. We would like the freedom to alter every word of that block independently of any other. The structure of M_N, and possibly of M_{N-1} , is constrained by Stage 1 and Stage 2 processing, these are therefore not easy blocks to attack. We can focus our attack on any other block or blocks. For purposes of this paper we will attack a single block $M_{a,a\neq N,N-1}$ and will hold all other blocks in M_1, ... , M_{N-2} identical in M and M^*.

2.2 Block Processing

The message digest register, *MD*, begins in a specified constant state MD_0 and is updated during the processing of each block. Its final state MD_N is the value assigned to MD5(*m*).

The processing of the *j*th block involves four *round functions*, *FF*, *GG*, *HH*, and *II*, as follows:

$$MD_j = MD_{j-1} + II\left(M_j, HH\left(M_j, GG\left(M_j, FF\left(M_j, MD_{j-1}\right)\right)\right)\right)$$

The round functions are similar to one another in structure. The message digest register is treated as a four-element shift register, with each element being one word wide. The elements are referred to as *A*, *B*, *C*, and *D*. Each round consists of 16 steps of this register.

At each step, $A = B + ((A + f(B,C,D) + x[s] + t) <<< k)$, where f is an *auxiliary function* which varies from round-to-round; $x[s]$ is a word chosen from M_j; s, t, and k are parameters of the step; and <<< k signifies a k-bit left circular shift of a word. Note that each step involves four + operations, one <<< operation, and one auxiliary function.

The auxiliary functions each take three 32-bit words as input and produce one 32-bit word as output. They are bit-wise parallel, which is to say that each bit of the output word depends only on the corresponding bits of the input words. The auxiliary functions f are defined in Table 1, where $\bar{v}$ denotes the bit-wise complement of v.

Table 1. Auxiliary functions.

Round	f	$f(X,Y,Z)$
FF	F	$XY \vee \bar{X}Z$
GG	G	$XZ \vee Y\bar{Z}$
HH	H	$X \oplus Y \oplus Z$
II	I	$Y \oplus (X \vee \bar{Z})$

3 The Cryptanalytic Problem

Message digest algorithms present two related cryptanalytic challenges. The simpler of these is to find two messages with the same digest. Rivest and Dusse conjecture that the difficulty of doing this for MD5 is on the order of 2^{64} operations. The other challenge is to find any message with a given digest. Rivest and Dusse conjecture that the difficulty of this feat for MD5 is on the order of 2^{128} operations.

We will attack the simpler problem, that of finding two messages $m \neq m^*$ such that MD5(m) = MD5(m^*). Note that under the simplistic assumption that the cryptanalytic difficulty of MD5 is uniformly distributed across its sixty-four steps, we will need to succeed at each step with probability > 0.5 in order to do better than Rivest's conjecture. This is a daunting prospect. Throughput is another measure of MD5's difficulty under differential cryptanalysis. MD5 produces only 2 output bits per step. DES produces 4.

The overwhelming number of + operations per step provides the motivation to attempt differential cryptanalysis mod 2^{32}. Analysis of the + operation is hard working mod 2; but it is easy working mod 2^{32}, and leads to output differences with probability = 1. On the other hand, analysis of the f functions is very hard mod 2^{32}. There is no theory to help, and a complete simulation is beyond the size of available computer memory. We must content ourselves with approximations. Table 2 summarizes the trade-offs facing the cryptanalyst.

Table 2. Cryptanalytic trade-offs within an MD5 step.

op	#	DCA mod 2	DCA mod 2^{32}
+	4	Hard analysis.	Easy analysis. See §3.1. Output differences with probability = 1.
<<<	1	Easy analysis. Output differences with probability = 1.	Non-trivial analysis. See §3.2. For any input difference only four output differences are possible, at least one has probability greater than or equal to 0.25.
f	1	Easy analysis. Low weight input differences lead to high probability output differences.	Hard analysis. See §3.3. Requires approximations. High probability output differences.

3.1 Differential Cryptanalysis of the Add Function

Let $x+c=z$, and $x^*+c=z^*$, then $z'=z-z^*=x-x^*=x'$, with probability = 1. Similarly, where $x+y=z$, and $x^*+y^*=z^*$, then $z'=z-z^*=x-x^*+y-y^*=x'+y'$, with probability = 1.

3.2 Differential Cryptanalysis of the Circular Shift

Theorem 1. If $i \in Z$ and $x \in R$, then $\lfloor i+x \rfloor = i + \lfloor x \rfloor$.

Theorem 2. If $a, b, m \in Z$, then

$$\left\lfloor \frac{a}{m} \right\rfloor - \left\lfloor \frac{a-b}{m} \right\rfloor = \begin{cases} \left\lfloor \frac{b}{m} \right\rfloor + 1, & a \bmod m < b \bmod m; \\ \left\lfloor \frac{b}{m} \right\rfloor, & a \bmod m \ge b \bmod m. \end{cases}$$

Let x be a 32-bit word and CLS$_k$(x) denote the circular left shift of x by k places. Let n=2^{32} and m=2^{32-k}. We can then write CLS$_k$(x) as the sum of two quantities

$$CLS_k(x) = 2^k x + \left\lfloor \frac{x}{m} \right\rfloor \bmod n.$$

The quantity $2^k x$ is simply a left shift of x by k places, with zeroes filled at the right. The quantity $\left\lfloor \frac{x}{m} \right\rfloor$ is just right shift of x by 32-k places, with zeroes filled at the left.

Recall that $x' = x - x^* \bmod n$, $0 \le x' \le n-1$. What does the mod n difference, $z' = CLS_k(x) - CLS_k(x^*) \bmod n$, look like? Assume $0 \le x, x' < n$. Then $z' = CLS_k(x) - CLS_k(x - x' \bmod n) \bmod n =$

$$2^k x + \left\lfloor \frac{x}{m} \right\rfloor - 2^k(x - x' \bmod n) - \left\lfloor \frac{x - x' \bmod n}{m} \right\rfloor \bmod n =$$
$$2^k x' + \left\lfloor \frac{x}{m} \right\rfloor - \left\lfloor \frac{x - x' \bmod n}{m} \right\rfloor \bmod n.$$

This can be divided, for further analysis, into two cases:

1. $x \ge x' \Rightarrow x - x' \ge 0 \Rightarrow x - x' \bmod n = x - x'$.
2. $x < x' \Rightarrow x - x' < 0 \Rightarrow x - x' \bmod n = x - x' + n$.

How many times does each case occur? If x' is held constant while x is varied over all possible n values, Case 1 occurs when x takes the $n - x'$ values $x = x', x'+1, \ldots, n-1$. Case 2 occurs when x takes the x' values $x = 0, 1, \ldots, x'-1$. We will consider the two cases separately.

Case 1. $x \ge x'$

Let $x = q_1 m + r_1, x' = q_2 m + r_2, 0 \le r_1, r_2 < m$. Then $\left\lfloor \frac{x}{m} \right\rfloor = q_1$, $\left\lfloor \frac{x'}{m} \right\rfloor = q_2$ and

$\left\lfloor \frac{x - x'}{m} \right\rfloor = q_1 - q_2 + \left\lfloor \frac{r_1 - r_2}{m} \right\rfloor$. By Theorem 2,

$$\left\lfloor \frac{x}{m} \right\rfloor - \left\lfloor \frac{x-x'}{m} \right\rfloor = \begin{cases} q_2, & \text{if } \left\lfloor \frac{r_1 - r_2}{m} \right\rfloor = 0; \\ q_2 + 1, & \text{if } \left\lfloor \frac{r_1 - r_2}{m} \right\rfloor = -1. \end{cases}$$

The quantity $\left\lfloor \frac{r_1 - r_2}{m} \right\rfloor = -1$when $r_1 < r_2$. How often does this occur? There are $2^k - q_2 - 1$ m-long sub-intervals in $[x', n\text{-}1]$. In each of these, $r_1 < r_2$ for the first r_2 values of x. So the event in question occurs $r_2(2^k - q_2 - 1)$ times.

Case 2. $x < x'$

Let x and x' be defined as in Case 1. We are now examining $\left\lfloor \frac{x}{m} \right\rfloor - \left\lfloor \frac{x-x'+n}{m} \right\rfloor \bmod n$ for $0 \le x < x < x' < n$. Note that $n/m = 2^k$, so

$$\left\lfloor \frac{x}{m} \right\rfloor - \left\lfloor \frac{x-x'+n}{m} \right\rfloor = \left\lfloor \frac{x}{m} \right\rfloor - \left\lfloor \frac{x-x'}{m} \right\rfloor - 2^k \bmod n =$$

$$\begin{cases} q_2 - 2^k, & \text{if } \left\lfloor \frac{r_1 - r_2}{m} \right\rfloor = 0; \\ q_2 - 2^k + 1, & \text{if } \left\lfloor \frac{r_1 - r_2}{m} \right\rfloor = -1. \end{cases}$$

We are again interested in how often $r_1 < r_2$ occurs. There are q_2 m-long sub-intervals in [0, x'-1]. In each of these, $r_1 < r_2$ for the first r_2 values of x. The final sub-interval is only r_2 long, and $r_1 < r_2$ throughout this final sub-interval. So the event in question occurs $r_2(q_2 + 1)$ times.

We have proven the following

Theorem 3. Select x and x^* at random from the integers in [0,n-1] so that $x' = x - x^* \bmod n$. Let $z' = CLS_k(x) - CLS_k(x^*) \bmod n$, and $x' = qm + r, 0 \le r < m$. Then

$$\Pr(z') = \frac{1}{n} \begin{cases} n - x' - r(2^k - 1 - q), & \text{if } z' = q + x'2^k \bmod n; \\ r(2^k - 1 - q), & \text{if } z' = q + 1 + x'2^k \bmod n; \\ r(q+1), & \text{if } z' = q - 2^k + 1 + x'2^k \bmod n; \\ x' - r(q+1), & \text{if } z' = q - 2^k + x'2^k \bmod n; \\ 0, & \text{otherwise.} \end{cases}$$

Note that this generalizes to other word sizes.

3.3 Differential Cryptanalysis of the Step Feedback Functions

The step feedback functions *f* present a difficult problem. In general, where $w = f(X,Y,Z)$ and $w^*=f(X^*,Y^*,Z^*)$, we are interested in $w' = w - w^* \bmod 2^{32} = f(X,Y,Z) - f(X^*,Y^*,Z^*) \bmod 2^{32}$.

A question which arises is, given X',Y', and Z', what can be said about w'? At present, nothing much is known either about the values of w' or their associated probabilities. We do not know how to do a mod 2^{32} theoretical analysis of these functions, and the memory required for a complete simulation is not available.

One hunch is that certain values of X',Y', and Z', for example those with low weight, may be likely to lead to values of w' with high probability. Another hunch is that it may be worthwhile to allow only one of X',Y', and Z' to take on a non-zero value. Monte Carlo evaluation within such restricted spaces begins to be feasible, and may lead to useful results.

In the meanwhile, we observe that the differential cryptanalyses mod 2^{32} and mod 2 are identical when we restrict X',Y', and Z' to differ in only the high-order bit. Analysis of the *f* s mod 2 is straightforward.

4 Cryptanalysis of the Rounds

We can now apply the analytic tools developed in §3 to the rounds of MD5. Recall that each round consists of 16 steps. Fig. 1 is a schematic representation of the calculations which make up a step. The constant additions are omitted as they make no difference from a differential point of view. The inputs have been rotated one element to the right for clarity. Table 3 contains data from an example analysis of the FF round. The notation used in Fig.1 and Table 3 is:

A, *B*, *C*, *D*: elements of the message digest shift register *MD*.

w: the output of the step auxiliary function *f*.

x: a word chosen from the current message block M_j. Specified in [RD].

z: the output of the circular left shift function CLS_k. [RD] specifies the values of *k*.

p, *q*: intermediate values included to clarify the illustration.

i: a step number. V_i indicates the value of variable *v* during step *i*. In the case of A_i, B_i, C_i, and D_i this is to be interpreted as the value of these message register elements at the conclusion of step *i*.

Our objective is to find two message blocks M_a and M_a^* such that $FF(M_a) = FF(M_a^*)$. We chose, arbitrarily, to work toward the end of the round. The situation at the end of step 11 is $A'_{11}, B'_{11}, C'_{11}, D'_{11} = 0$. Our objective will be met if we can introduce some difference, and then remove it, so that $A'_{16}, B'_{16}, C'_{16}, D'_{16} = 0$.

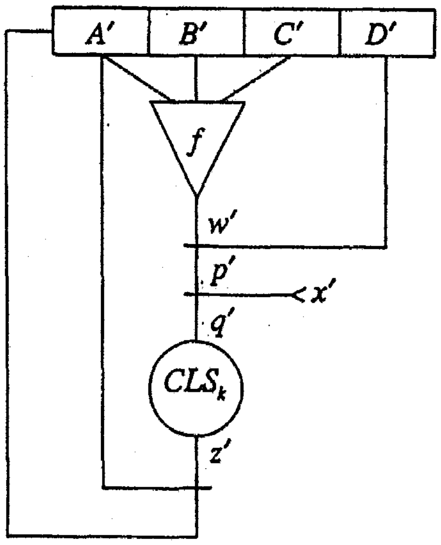

Figure 1. Schematic of an MD5 step, showing only those operations which impact the differential analysis.

Step 12. We can analyze the *FF* round auxiliary function F only in the case that A', B', and C' are restricted to the high order bit. So our immediate goal is to choose x'_{12} which leads to $A'_{12} = 2^{31}$. Examination of Fig. 1 and Table 3 shows that this reduces to choosing an x' such that $2^{31} = CLS_{22}(x')$. An important insight is that $z = CLS_k(x)$ is a permutation, whose inverse is $x = CLS_{n-k}(z)$. So it is possible to use the theory we developed in §3.2 to calculate the x' we are looking for. There are two values with high probability, we choose 2^9, whose probability is 1/2.

Step 13. We would now like to know the possible values and corresponding probabilities of $w' = F(X,Y,Z) - F(X^*,Y,Z)$. Carries have no effect when input differences are restricted to the high-order bit. The step auxiliary functions can then be evaluated as Boolean functions on a single bit. This straightforward, either symbolically or by enumeration. For the function at hand, the h.o. bit of w' = 0 or 1, each with probability of 1/2. We choose the 0. Now our goal is to choose an x'_{13} such that $A'_{13} = 0 \Rightarrow z'_{13} = 2^{31}$. Working with the inverse of CLS_7 we choose $x' = 2^{24} \Rightarrow z'_{13} = 2^{31}$ with probability 1/2.

Table 3. Step-by-step analysis of round *FF*.

i	k_i	w'_i	p'_i	x'_i	q'_i	z'_i	prob	A'_i	B'_i	C'_i	D'_i
11							1	0	0	0	0
12	22	0	0	2^9	2^9	2^{31}	$1 \cdot \frac{1}{2}$	2^{31}	0	0	0
13	7	0	0	2^{24}	2^{24}	2^{31}	$\frac{1}{2} \cdot \frac{1}{2}$	0	2^{31}	0	0
14	12	0	0	0	0	0	$\frac{1}{2} \cdot 1$	0	0	2^{31}	0
15	17	0	0	0	0	0	$\frac{1}{2} \cdot 1$	0	0	0	2^{31}
16	22	0	2^{31}	2^{31}	0	0	$1 \cdot 1$	0	0	0	0
							$\prod = 2^{-5}$				

Step 14. Now we need to know the values and probabilities of $w' = F(X,Y,Z) - F(X,Y^*,Z)$. Working as in Step 12, we calculate the h.o. bit of w' = 0 or 1, each with probability of 1/2. Again we choose the 0. Our goal is now to choose an x'_{14} such that $A'_{14} = 0 \Rightarrow z'_{14} = 0$. Thus x'_{14} is trivially 0, and leads to z'_{14}=0 with probability 1.

Step 15. Now we need to know the values and probabilities of $w' = F(X,Y,Z) - F(X,Y,Z^*)$. Working as in Step 12, we calculate the h.o. bit of w' = 0 or 1, each with probability of 1/2. Again we choose the 0. Our goal is now to choose an x'_{15} such that $A'_{15} = 0 \Rightarrow z'_{15} = 0$. As in the previous step, x'_{15} is trivially 0, which leads to $z'_{15} = 0$ with a probability of 1.

Step 16. By inspection, w' = 0 with probability 1. Our goal is to choose an x'_{16} such that A'_{16} = 0. Notice that $p'_{16} = 2^{31}$. If we select $x'_{16} = 2^{31}$ then $q'_{16} = 0$, since all additions are mod 2^{32}. Since $q'_{16} = 0$ then $A'_{16} = 0$ with probability 1. We have reached our goal: $A'_{16}, B'_{16}, C'_{16}, D'_{16} = 0$. This completes the analysis of round *FF*.

The procedure just described has also been applied to rounds *GG*, *HH*, and *II*. The results for all four rounds are summarized in Table 4.

It should be clear that each time one of the equiprobable w's was chosen, we could have as easily chosen the other, which would have lead to a different differential. What may be less clear is that the second most probable z' often occurs with probability 1/2-ε, and also leads to useful differentials. A last comment is that the differentials exhibited here can be slid as far forward in a round as the round's first step.

Table 4. Example message block differentials M'_a for which $roundfunction(M'_a, MD) = MD$.

word	*FF*	*GG*	*HH*	*II*
0	0	0	0	0
1	0	0	0	0
2	0	0	2^{31}	0
3	0	0	0	0
4	0	0	0	2^{25}
5	0	0	0	0
6	0	0	$2^{32} - 2^{8}$	0
7	0	0	0	0
8	0	2^{11}	0	0
9	0	0	$78\,00\,00\,00_x$	2^{31}
10	0	0	0	0
11	2^{9}	0	0	2^{31}
12	2^{24}	2^{31}	2^{31}	0
13	0	2^{26}	0	2^{10}
14	0	0	0	0
15	2^{31}	0	2^{31}	0
probability	2^{-5}	2^{-2}	2^{-2}	2^{-4}

The challenge remains to find a single M'_a which makes all four rounds of MD5 ineffective simultaneously.

References

[BKPS] Lawrence Brown, Matthew Kwan, Josef Pieprzyk and Jennifer Seberry, "Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI," in *Asiacrypt '91 Abstracts*, pp. 25-30.

[BS1] Eli Biham and Adi Shamir, "Differential Analysis of DES-like Cryptosystems," in *Advances in Cryptology -- Crypto '90*, pp. 2-21.

[BS2] Eli Biham and Adi Shamir, "Differential Analysis of DES-like Cryptosystems," *Journal of Cryptology* (1991) 4:1, pp. 3-72.

[BS3] Eli Biham and Adi Shamir, "Differential Analysis of FEAL and N-Hash," in *Advances in Cryptology -- Eurocrypt '91*, pp. 1-16.

[BS4] Eli Biham and Adi Shamir, "Differential Analysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer," in *Advances in Cryptology -- Crypto '91.*

[Knud] Lars Ramkilde Knudsen, "Cryptanalysis of LOKI," in *Asiacrypt '91 Abstracts*, pp. 19-24.

[LMM] Xeujia Lai, James L. Massey and Sean Murphey, "Markov Ciphers and Differential Cryptanalysis," in *Advances in Cryptology -- Eurocrypt '91*, pp. 17-38.

[RD] R. Rivest and S. Dusse, "The MD5 Message-Digest Algorithm," Network Working Group Internet Draft, RSA Data Security Inc., 10 July 1991.

[Riv] Ronald Rivest, "MD5", presentation at Crypto '91 rump session.