Aumentando a segurança do MD6 em relação aos ataques diferenciais

Valdson S. Cleto[1], **Routo Terada**[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
São Paulo – SP – Brazil

vcleto@gmail.com, rt@ime.usp.br

Abstract. *This paper proposes a modification on the compression function of the MD6 hash function that increases the security of the function regarding the differential attacks. Such modification enables a reduction of up to 28% in the number of rounds needed to demonstrate the strength of the MD6 compression function against differential attacks.*

Resumo. *Este artigo propõe uma modificação na função de compressão da função de hash MD6 para aumentar a segurança da função em relação aos ataques diferenciais. Tal modificação possibilita uma redução de até 28% no número de rodadas necessárias para a demonstração da resistência da função de compressão do MD6 aos ataques diferenciais.*

1. Introdução

A função de hash MD6 foi apresentada em outubro de 2008 por [Rivest et al. 2008] como uma candidata para a competição organizada pelo instituto norte-americano NIST (National Institute of Standards and Technology) para a escolha de um novo algoritmo de hash padrão, que receberá o título de SHA-3 (o algoritmo padrão de hash atual é o SHA-2).

Porém, em julho de 2009 Ron Rivest emitiu um comunicado (http://groups.csail.mit.edu/cis/md6/OFFICIAL_COMMENT_MD6_2009-07-01.txt) informando que naquele momento o MD6 não atenderia os requisitos de velocidade necessários para um candidato a SHA-3 e portanto não recomendava que o MD6 passasse para a segunda fase da competição. Então o MD6 não apareceu na lista dos candidatos que passaram à segunda fase.

O NIST estabeleceu que, para ser competitivo, um candidato a SHA-3 precisaria ser no mínimo tão rápido quanto o SHA-2 em plataformas de referência. Embora o MD6 seja muito rápido em sistemas multiprocessados, nas plataformas de referência ele é bem mais lento que o SHA-2.

Ron Rivest alertou os organizadores da competição que o algoritmo de SHA-3 que viesse a ser escolhido deveria ser demonstravelmente resistente a ataques diferenciais, visto que foi o poder surpreendente dos ataques diferenciais que estimulou a competição para escolha do SHA-3.

O que torna o MD6 significativamente mais lento que os outros competidores nas plataformas de referência é o número de rodadas da função de compressão que teve que ser adotado justamente para torná-lo demonstravelmente resistente a ataques diferenciais.

A demonstração da resistência do MD6 a ataques diferenciais é apresentada na seção 6.9 em [Rivest et al. 2008]. Ao final dessa seção são sugeridas algumas possibilidades de investigação para se tentar demonstrar a resistência do MD6 a ataques diferencias com um número menor de rodadas. O resultado da investigação de uma dessas possibilidades foi a descoberta de uma modificação na função de compressão do MD6 que permite que a demonstração da resistência do MD6 a ataques diferencias seja feita com uma redução de até 28% no número de rodadas, o que resulta na possibilidade de aumentar a velocidade de processamento do MD6 praticamente nesta mesma proporção.

2. Demonstração da resistencia do MD6 a ataques diferenciais

A investigação apresentada nesse artigo foi feita a partir da demonstração da resistência do MD6 a ataques diferenciais apresentada em [Rivest et al. 2008] na seção 6.9. Nesta seção será apresentada uma visão geral dessa demonstração.

Para a demonstração é feita uma análise da resistência da função de compressão do MD6 a ataques diferenciais que buscam encontrar uma colisão na função de hash. Estes ataques consistem em se escolher pares de mensagens de entrada com determinadas diferenças tentando-se encontrar um par tal que o par de resumo da mensagem na saída da função de hash não tenha diferença, o que significa encontrar uma colisão. Se a probabilidade de se encontrar esse par de mensagens não é desprezível, então calculando-se o resumo das mensagens de uma quantidade suficiente de pares de mensagens de entrada pode-se encontrar uma colisão.

A função de compressão do MD6 pode ser representada pelo algoritmo 1.

Algoritmo 1 Função de compressão

Entrada: $A[0 \cdots 88]$ de $A[0 \cdots 16r + 88]$

para $i = 89$ **a** $16r + 88$ **:**

$x = S_i \oplus A[i - 17] \oplus A[i - 89] \oplus (A[i - 18] \wedge A[i - 21]) \oplus (A[i - 31] \wedge A[i - 67])$

$x = x \oplus (x >> r_i)$

$A[i] = x \oplus (x << l_i)$

retorne $A[16r + 73 \cdots 16r + 88]$

No algoritmo 1, $A[0..88]$ é o vetor com as 89 palavras de entrada. r é o número de rodadas. A cada rodada são calculadas $c = 16$ novas palavras. $A[0..16r + 88]$ é o vetor completo com as 89 palavras de entrada mais as $t = 16r$ palavras calculadas nas r rodadas. Cada palavra é calculada a partir das 89 palavras imediatamente anteriores a ela no vetor. As 16 palavras calculadas na última rodada são a saída da função de compressão.

As escolhas dos índices relativos 17, 18, 21, 31 e 67 visam otimizar a difusão. As constantes S_i mudam ao final de cada rodada. As quantidades de deslocamento de bits se repetem a cada rodada e são definidas pela tabela 1, que também visa à obtenção da difusão máxima.

Para a demonstração da resistência da função de compressão a ataques diferenciais, antes de mais nada deve-se estabelecer uma forma de medir a diferença entre duas mensagens e esta forma pode variar de acordo com as operações envolvidas na função

Tabela 1. Quantidade de deslocamento de bits

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r_i	10	5	13	10	11	12	2	7	14	15	7	13	11	7	6	12
l_i	11	24	9	16	15	9	27	15	6	2	29	8	15	5	31	9

de hash. A forma de medida mais utilizada é o ou-exclusivo, e é a forma utilizada nessa demonstração.

Um *caminho diferencial* é um conjunto de diferenças entre o par de entradas, todos os estados intermediários e o par de saídas. Para o MD6 podemos expressar um caminho diferencial como: ΔA_i para $i = 0, \ldots, t + n - 1$.

É fácil notar que um *caminho diferencial de colisão* é um caminho onde $\Delta A_i = 0$ para $i = t + n - c, \ldots, t + n - 1$.

A propriedade mais importante de um caminho diferencial é a sua *probablidade* associada. A probabilidade de um determinado passo i de um caminho diferencial, p_i, é definida como a probabilidade de que o par de saída do passo siga o caminho diferencial, dado que o par de entrada satisfaz a diferença especificada pelo caminho diferencial.

A probabilidade total de um caminho diferencial, p, é o produto das probabilidade em todos os passos, se for assumido que o cálculo dos passos são independentes entre si.

Definimos D_i como o peso de Hamming de uma determinada diferença ΔA_i, ou seja, o número de bits diferentes entre A_i e A'_i, ou $D_i = |\Delta A_i|$. Então, para um caminho diferencial $\{\Delta A_i\}$ definimos um *caminho diferencial de padrão de peso* como $\{D_i\}$.

2.1. Análise das propriedades diferencias das operações da função de compressão

Cada rodada da função de compressão do MD6 é composta por 16 passos e em cada passo uma nova palavra de 64 bits é calculada Para a análise das propriedades diferenciais de cada uma das 3 diferentes operações contidas em cada passo: XOR, AND e o operador g, que representa as operações de deslocamentos de bits, será adotada a seguinte notação:

- X, Y, Z para as entradas e saídas de w bits
- $\Delta X, \Delta Y, \Delta Z$ para as diferenças
- D_X, D_Y, D_Z para os pesos de Hamming
- x, y, z para um bit de palavras de w bits

A propriedade diferencial da operação XOR é direta, $\Delta Z = \Delta X \oplus \Delta Y$. Em termos do peso de Hamming, temos que: $\max(D_X, D_Y) - \min(D_X, D_Y) \leq D_Z \leq D_X + D_Y$.

Uma operação AND entre duas paravras de w bits pode ser vista como um conjunto de w portas AND independentes. Se os bits de entrada de cada porta AND forem x e y e a saída for z, o comportamento diferencial da porta AND depende das diferenças nas entradas, ou seja, Δx e Δy. Consideramos estes dois casos:

- Chamamos de porta AND "inativa" quando $\Delta x = \Delta y = 0$ e portanto temos que $\Pr[\Delta z = 0] = 1$.
- Chamamos de porta AND "ativa" quando $\Delta x = 1$ ou $\Delta y = 1$ e portanto temos que $\Pr[\Delta z = 0] = \Pr[\Delta z = 1] = 1/2$

Em termos do peso de Hamming, temos que:

$$0 \leq D_Z \leq D_X + D_Y \tag{1}$$

As portas AND ativas, ou AAG's, do inglês, Active AND Gates, serão fundamentais na demonstração da carga de trabalho mínima de um ataque diferencial, já que esta é a única operação não trivial em termos de probabilidades diferenciais. Uma porta AND ativa (AAG) sempre contribui com um probabilidade igual a $1/2$ para a probabilidade total do caminho diferencial, não importa qual seja a diferença de saída da porta AND. O número total de portas AND ativas em um caminho diferencial está diretamente relacionado à probabilidade total do caminho.

O operador $g_{r,l}$ faz um espalhamento dos bits dentro de uma palavra. Sabemos que se $Z = g_{r,l}(X)$, então $\Delta Z = g_{r,l}(\Delta X)$.

A combinação de um deslocamento e um XOR pode no máximo dobrar o número de diferenças, como são realizadas duas combinações de operações (uma com deslocamento pra direita e outra com deslocamento pra esquerda) temos que: $D_Z \leq 4D_X$.

Cada par de quantidade de deslocamentos (r, l) foi escolhido de forma que se $0 < D_X \leq 4$ então $D_Z \geq 2$.

Ou seja, para que a diferença na saída seja de apenas um bit é necessário que a diferença na entrada seja de 5 ou mais bits. Isto foi projetado desta forma para impedir a propagação de diferenças de apenas um bit, dificultando a obtenção de caminhos diferenciais com pesos de Hamming muito baixos, sendo impossivel conseguir um caminho onde todos os pesos são no máximo 1.

Se $D_X > 4$ então $D_Z > 0$, já que se existem diferenças na entrada devem existir diferenças na saída.

Vamos agora combinar em duas partes as operações executadas em um passo:

$$X = A_{i-t_0} \oplus A_{i-t_5} \oplus (A_{i-t_1} \wedge A_{i-t_2}) \oplus (A_{i-t_3} \wedge A_{i-t_4}), \tag{2}$$

$$A_i = g(X). \tag{3}$$

Usando as desigualdadas apresentadas para cada operação podemos derivar limites superior e inferior para D_X:

$$D_X \leq UB_X = \sum_{k=0}^{5} D_{i-t_k}, \tag{4}$$

$$D_X \geq LB_X = \max(D_{i-t_0}, D_{i-t_5}) - \min(D_{i-t_0}, D_{i-t_5}) \sum_{k=1}^{4} D_{i-t_k}. \tag{5}$$

Focando no peso de Hamming ao invés de se focar no real valor das diferenças perde-se certa precisão na análise, mas evita-se a complicação de ter que analizar como as diferenças de bit individualmente podem se alinhar de um operação para outra, além de possibilitar a busca de caminhos diferenciais de padrões de peso válidos através de uma busca auxiliada por um programa computacional.

2.2. Carga mínima de trabalho de um ataque diferencial padrão

O objetivo agora é provar que ataques diferenciais padrão contra o MD6 são menos eficientes para encontrar colisões do que o ataque pelo paradoxo de aniversário. Ou seja, precisamos provar que a probabilidade de se encontrar qualquer caminho diferencial de colisão na função de compressão do MD6 é no máximo $2^{-d/2}$, o que significa dizer que a carga de trabalho de um ataque diferencial padrão é no mínimo $2^{d/2}$, que é o limite teórico do paradoxo do aniversário.

Como vimos, cada porta AND ativa em um caminho diferencial contribui com a probabilidade de 1/2, então se o número de portas AND ativas em um caminho diferencial válido do MD6 é no mínimo $d/2$, a probabilidade associada a este caminho será no máximo $2^{-d/2}$.

Cada diferença de bit em um caminho diferencial de padrões de peso pode ativar até 4 portas AND em 4 passos distintos, uma para cada posição t_1, t_2, t_3 e t_4. Em alguns casos uma diferença de bit pode não ativar as 4 portas AND, e estes casos devem ser levados em consideração para não contarmos portas AND ativas a mais:

- Se duas diferenças de bit ativam a mesma porta AND.
- Se duas portas AND são ativadas no mesmo passo.
- Se uma porta AND está além do limite de rodadas. Só contamos as portas AND ativas que tem as duas entradas dentro do limite de rodadas em que está sendo feita a busca.

Para fazer a busca de caminhos diferenciais de padrões de peso possíveis desejamos eliminar o máximo possível de padrões inválidos. Utilizando (4) e (5) e as desigualdades mostradas para a função g, podemos eliminar os seguintes valores de D_i em um determinado passo i:

1. $D_i = 0$ e $LB_X > 0$
2. $D_i > 4UB_X$
3. $D_i = 1$ e $UB_X < 5$

A tabela 2 mostra o resultado apresentado em [Rivest et al. 2008], obtido através de um programa computacional para buscar a número mínimo de portas AND ativas em qualquer padrão de peso de caminho diferencial de até s rodadas.

Tabela 2. Número mínimo de portas AND ativas em qualquer padrão de peso de caminho diferencial de até s rodadas

s	≤ 5	6	7	8	9	10	11	12	13	14	15
Número mínimo de portas AND ativas	0	3	4	4	4	4	7	13	19	20	26

Os valores encontrados na tabela 2 (principalmente o valor do número mínimo de portas AND ativas em $s = 15$ rodadas, 26) podem ser utilizados para expandir o resultado a um número r qualquer de rodadas através da fórmula: $AAG_r \geq AAG_s \times \lfloor r/s \rfloor$, onde AAG_x é o número mínimo de portas AND ativas em x rodadas ($AAG =$ Active AND Gate).

Antes disso deve-se tomar o cuidado de deixar uma margem de segurança, porque alguém que tente atacar a função pode conseguir penetrar algumas rodadas no começo do

cálculo do hash manipulando as entradas e influenciando o comportamento do caminho diferencial. Estabeleceu-se uma margem de segurança conservadora de 15 rodadas, ou seja, substitui-se na fórmula o número de rodadas r por $r - 15$.

A tabela 3 mostra o resultado apresentado em [Rivest et al. 2008] para a carga de trabalho mínima de um ataque diferencial padrão ao MD6 comparada com a carga de trabalho de um ataque pelo paradoxo do aniversário, mostrando que a carga de trabalho de um ataque diferencial é maior que a carga de trabalho de um ataque pelo paradoxo do aniversário, que é o que se desejava demonstrar.

Tabela 3. Resultado apresentado em [Rivest et al. 2008] para a carga de trabalho mínima de um ataque diferencial padrão ao MD6 (LB é a carga de trabalho mínima e BB é a carga de trabalho de um ataque pelo paradoxo do aniversário)

d	r	$r-15$	$\lfloor \frac{r-15}{15} \rfloor$	$AAG_{r-15} \geq$	$LB \geq$	BB
40	50	35	2	52	2^{52}	2^{20}
80	60	45	3	78	2^{78}	2^{40}
128	72	57	3	78	2^{78}	2^{64}
160	80	65	4	104	2^{104}	2^{80}
224	96	81	5	130	2^{130}	2^{112}
256	104	89	5	150	2^{150}	2^{128}
384	136	121	8	208	2^{208}	2^{192}
512	168	153	10	260	2^{260}	2^{256}

3. Redução do número de rodadas necessárias para a demonstração da resistência a ataques diferenciais

Até aqui mostramos os resultados apresentados em [Rivest et al. 2008], nesta seção mostraremos os resultados de nossa investigação.

Ao apresentar a demonstração da segurança do MD6 contra ataques diferenciais padrão, mostramos que ela é dependente do número de rodadas utilizado na função de compressão. O número de rodadas deve garantir uma quantidade mínima de portas AND ativas na execução da função de compressão pois a resistência a um ataque diferencial está diretamente relacionada a essa quantidade.

Ao final da seção 6.9.3.4 de [Rivest et al. 2008] são apresentadas algumas possibilidades de investigação para se tentar demonstrar que o número mínimo de portas AND ativas em um número reduzido de rodadas s é maior do que o encontrado. Uma dessas possibilidades diz que podem não existir caminhos diferenciais válidos para alguns dos padrões de peso de caminho diferencial encontrados.

Investigamos a existência de caminhos diferenciais válidos para cada padrão de peso de caminho diferencial encontrado. Para isso, implementamos um algoritmo para realizar a busca por padrões de peso de caminho diferencial de forma a obtermos os mesmos resultados apresentados na seção anterior. Então, acrescentamos a essa implementação um código para a busca por caminhos diferenciais válidos para um dado padrão de peso de caminho diferencial.

Encontramos caminhos diferenciais válidos e, ao analisarmos como esses cami-

nhos se formavam, identificamos algumas características da tabela de quantidade de deslocamento de bits (tabela 1) que possibilitavam a formação desses caminhos diferenciais.

Então, modificamos o programa de busca da tabela de quantidade de deslocamento de bits utilizado pelos autores do MD6 para a definição da tabela 1. Esse modificação foi feita para que a busca procurasse por tabelas sem as características que identificamos como as responsáveis pela formação dos caminhos diferenciais válidos para os padrões de peso de caminho diferencial com número mínimo de portas AND antivas. Encontramos uma nova tabela de acordo com essa restrição.

Os resultados serão apresentados nesta seção.

3.1. Verificando a existência de caminhos diferenciais válidos

O padrão de peso de caminho diferencial encontrado que resulta no número mínimo de portas AND ativas em s rodadas pode não corresponder a nenhum caminho diferencial válido. Por isso, adicionamos ao programa de busca de padrões de peso de caminho diferencial a busca por um caminho diferencial válido quando um padrão de peso de caminho diferencial é encontrado. Caso nenhum caminho diferencial seja encontrado a busca por um padrão de peso de caminho diferencial continua enquanto não for encontrado um caminho diferencial válido que corresponda a um dado padrão de peso de caminho diferencial.

Para 15 rodadas, vimos que o número mínimo de portas AND ativas é 26. Nosso programa deve procurar algum caminho diferencial válido correspondente ao padrão de peso de caminho diferencial encontrado, ou a qualquer outro padrão de peso de caminho diferencial com 26 portas AND ativas.

Para buscar um caminho diferencial válido testamos todas as possibilidades de valores diferenciais possíveis para cada valor de peso do padrão de peso de caminho diferencial e utilizamos as propriedades de cada uma das operações da função de compressão conforme mostrado em 2.1 na página 3.

Com este programa, descobrimos que para o primeiro padrão de peso de caminho diferencial encontrado para $s = 15$ com 26 portas AND ativas ($D_{54} = 1, D_{71} = 2, D_{143} = 2, D_{232} = 2$), não existe caminho diferencial válido. Mas o programa encontrou outros padrões de peso de caminho diferencial com 26 portas AND ativas, e encontrou um que tem um respectivo caminho diferencial válido. Para este padrão de peso de caminho diferencial: $D_{28} = 2, D_{83} = 1, D_{100} = 2, D_{172} = 2$ existe um caminho diferencial válido: $A_{28} = 0\text{x}8001, A_{83} = 0\text{x}1, A_{100} = 0\text{x}8001, A_{172} = 0\text{x}8001$ (valores em hexadecimal).

3.2. Análise dos caminhos diferenciais válidos encontrados

Analisando o caminho diferencial com 26 portas AND ativas em 15 rodadas ($A_{28} = 0\text{x}8001, A_{83} = 0\text{x}1, A_{100} = 0\text{x}8001, A_{172} = 0\text{x}8001$) vemos que a formação dele é possível porque os valores de deslocamento de bits no passo 4 de uma rodada são iguais aos valores de deslocamento de bits do passo 12, como mostrado na tabela de deslocamento de bits 1: 11 bits para a direita e 15 bits para a esquerda. A diferença entre as posições t_0 e t_5 módulo 16 é igual a 8 (89 - 17 = 72; 72 módulo 16 = 8), ou seja, é igual à diferença entre as posições da tabela de deslocamento de bits que contém o mesmo valor de deslocamento de bits, as posições 4 e 12. Assim, um valor diferencial que apareça na

posição t_0 em um passo 4 ou 12 no módulo 16, necessariamente aparecerá na posição t_5 em um passo 12 ou 4 no módulo 16, respectivamente, e a este valor diferencial será aplicado o mesmo deslocamento de bits, resultando no alinhamento e cancelamento destes valores diferenciais em um passo posterior. No caminho diferencial encontrado, o valor diferencial do passo 83 aparece na posição t_0 do passo 100, e 100 módulo 16 é igual a 4. É também o valor diferencial na posição t_5 do passo 172, e 172 módulo 16 é igual a 12. Portanto nos passos 100 e 172 obtemos o mesmo valor diferencial por que é aplicado nesse passos o mesmo delocamento de bits ao valor diferencial do passo 83. No passo 189, o valor diferencial gerado no passo 100 estará na posição t_5 e o valor diferencial do passo 172 estará na posição t_0. Como eles são iguais, ocorre um alinhamento das diferenças e elas são anuladas.

Concluímos que esta coincidência de valores da tabela de deslocamento de bits 1 a uma distância que coincide com a diferença entre as posições t_0 e t_5 no módulo 16 é uma falha na escolha dos valores de deslocamento de bits. Seria interessante tentar escolher uma outra tabela onde esta coincidência não ocorra, verificando como a função se comporta com esta alteração

3.3. Investigando uma nova tabela de deslocamento de bits

A escolha da tabela de deslocamento de bits 1 foi feita através de um programa computacional disponibilizado pelos autores do MD6. Este programa procura uma tabela de deslocamento tentando maximizar a taxa de difusão dos bits dentro das palavras, dadas as posições t_0 a t_5 e estabelecidas algumas exigências na escolha dos valores de deslocamento. Cada valor de deslocamento não pode ser zero, deve ser no máximo $w/2$ (32) e r_i e l_i não devem ser múltiplos um do outro. Cada par de valores (r_i, l_i) deve ser escolhido tal que uma saída com peso de hamming igual a 1 não possa ser gerado por uma palavra de entrada de peso menor que cinco. Além disso, r_i e l_j não podem ser múltiplos um do outro para qualquer j tal que $(i - j) \in t_0, t_5, t_5 - t_0$ (todos os índices no módulo $c = 16$). Estas últimas condições ajudam a garantir que um deslocamento à esquerda em uma rodada não será seguido por um deslocamento à direita pela mesma quantidade (ou um múltiplo) em uma rodada posterior. Para cada tabela gerada aleatoriamente de acordo com as restrições descritas é medido um valor para que as tabelas possam ser comparadas de forma que seja escolhida a tabela que garanta o efeito avalanche mais rápido entre as tabelas testadas. Para a escolha da tabela de deslocamento de bits original do MD6 foram testadas 1 milhão de tabelas.

Como mostramos, notamos que outras condições poderiam ser impostas aos valores da tabela de deslocamento de bits para evitar a formação de alguns caminhos diferenciais com baixos valores de peso de hamming. Fomos então adicionando novas restrições aos valores da tabela, procurando novas tabelas, testando a nova tabela encontrada e descobrindo novas restrições que poderiam ser impostas. Eliminamos todas as características da tabela de deslocamento de bits que contribuiam para a formação de caminhos diferenciais com 26 portas AND ativas em 15 rodadas. Ainda assim, existe caminho diferencial com 26 portas AND ativas em 15 rodadas, mas esse caminho não depende de nenhuma característica especial da tabela de deslocamento de bits, ou seja, nenhuma tabela de deslocamento de bits evitaria a formação desse caminho.

As restrições adicionais que descobrimos que devem ser impostas para que a ta-

bela de deslocamento de bits não possibilite a formação de alguns dos caminhos diferenciais com 26 portas AND ativas em 15 rodadas estão descritas a seguir:

1. Se $i - j = t_5 - t_0$ módulo c, então:
   l_i deve ser diferente de l_j e
   r_i deve ser diferente de r_j se $l_j > r_j$ e $l_i > r_i$.
2. Se $i - j = t_5$ módulo c, então:
   l_i deve ser diferente de l_j se $r_i > l_i$ e
   r_i deve ser diferente de r_j se $l_j > r_j$ e $l_i > 2r_i$.

Essas restrições foram implementadas na função que gera tabelas aleatórias de deslocamento de bits. Esta função faz parte do código fornecido pelos autores do MD6 que foi utilizado para a busca da tabela de deslocamento de bits original do MD6.

Executando o programa de busca da tabela de deslocamento de bits com essas restrições adicionais e testando a mesma quantidade de tabelas que foram testadas para a escolha da tabela original do MD6, 1 milhão de tabelas, a melhor tabela encontrada é um pouco pior do que a tabela original do MD6, de acordo com a medida usada para a comparação das tabelas, que é uma medida da taxa de difusão dos bits dentro das palavras obtida pela tabela. Continuando a busca, foi encontrada uma tabela melhor do que a tabela original do MD6 de acordo com essa medida da taxa de difusão de bits (e que ainda atende às restrições que adicionamos).

O programa de busca utiliza uma semente para a geração de uma tabela de deslocamento de bits aleatória. Ele começa a busca com a semente 0, e vai incrementando esse valor. Então, para 1 milhão de tabelas testadas, a semente utilizada para a geração da última tabela é igual a 999.999. A tabela original do MD6 foi gerada com a semente 939.663. Com as restrições adicionais, encontramos a melhor tabela ao testar a semente número $1.421.812$. Os resultado obtidos podem ser rapidamente verificados com o programa fornecido pelos autores do MD6 (shiftopt.c), os valores das sementes que geram as melhores tabelas e as alterações no código que implementam as restrições adicionais descritas acima. A tabela de deslocamento de bits que encontramos é a tabela 4.

Tabela 4. Nova tabela de deslocamento de bits encontrada: melhor taxa de difusão de bits em relação à tabela original do MD6 e atende às restrições adicionais para impedir a formação de alguns dos caminhos diferenciais com 26 portas AND ativas em 15 rodadas.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
r_i	13	13	7	8	11	9	10	4	11	14	2	12	11	8	6	12
l_i	4	9	23	10	5	21	13	18	12	3	27	7	15	17	23	5

3.4. Resultados obtidos com a nova tabela de deslocamento de bits

Com a tabela de deslocamento de bits original do MD6, o primeiro padrão de peso de caminho diferencial com 26 portas AND ativas em 15 rodadas que possui um caminho diferencial válido correspondente encontrado pelo programa de busca é:

$$D_{28} = 2, D_{83} = 1, D_{100} = 2, D_{172} = 2. \quad (6)$$

Com a nova tabela de deslocamento de bits não existe um caminho diferencial válido para este padrão de peso de caminho diferencial. Mas, com qualquer tabela de deslocamento

de bits, existe um outro padrão de peso de caminho diferencial com 26 portas AND ativas em 15 rodadas que possui um correspondente caminho diferencial válido:

$$D_{16} = 2, D_{71} = 1, D_{88} = 2, D_{160} = 2. \quad (7)$$

Podemos observar que estes dois padrões de peso de caminho diferencial são semelhantes, estando apenas deslocados de 12 posições. O que possibilita a existência de um caminho diferencial válido correspondente ao padrão de peso de caminho diferencial (7), independente da tabela de deslocamento de bits usada, é o fato do índice 88 fazer parte das primeiras 89 palavras do caminho, e portanto o valor diferencial desta posição depende de um valor diferencial que não faz parte do caminho, que é anterior ao valor diferencial da posição de índice 0. Sendo assim, o valor diferencial da posição 88 depende de um valor diferencial desconhecido que consideramos que possa ser qualquer valor. No padrão de peso de caminho diferencial (6), para que os valores diferenciais se anulem na posição 189, é necessário que o valor diferencial da posição 83 resulte nos mesmos valores diferenciais nas posições 100 e 172. Já no padrão de peso de caminho diferencial (7), como o valor diferencial da posição 88 não depende apenas do valor diferencial da posição 71, mas também de um valor desconhecido, o valor diferencial da posição 88 pode ser igual ao valor diferencial da posição 160 independente da tabela de deslocamento de bits usada.

A vantagem da nova tabela de deslocamento de bits aparece quando buscamos caminhos diferenciais válidos em 16 rodadas. Esse deslocamento de 12 posições entre (6) e (7) faz muita diferença quando 1 rodada é adicionada ao cálculo. O valor diferencial da posição 172 em (6) aparecerá no cálculo do valor diferencial da posição 261 ($172+89$). Mas, em 16 rodadas temos 256 passos, portanto a posição 261 está além do cálculo de 16 rodadas. Desta forma, com a tabela de deslocamento de bits original do MD6, o mesmo caminho diferencial com 26 portas AND ativas em 15 rodadas correspondente ao padrão de peso de caminho diferencial (6) é válido para 16 rodadas. Já o valor diferencial da posição 160 em (7) aparecerá no cálculo do valor diferencial da posição 249 ($160 + 89$), que está dentro do cálculo de 16 rodadas.

Executando o programa de busca para 16 rodadas com a nova tabela de deslocamento de bits, comprovamos a existência de um caminho diferencial válido para o seguinte padrão de peso de caminho diferencial:

$$D_{16} = 2, D_{71} = 1, D_{88} = 2, D_{160} = 2, D_{249} = 4. \quad (8)$$

Este padrão de peso de caminho diferencial é uma extensão a 16 rodadas de (7). O número de portas AND ativas neste caminho é 38.

A busca por padrões de peso de caminho diferencial com 26 portas AND ativas ou mais é bem demorada. Até o momento conseguimos comprovar que com a nova tabela não existem caminhos diferenciais válidos com até 27 portas AND ativas. Não conseguimos comprovar a inexistência de caminhos diferenciais válidos com mais de 27 e menos do que 38 portas AND ativas. Pelo que temos observado dos resultados da busca computacional e pelo que conseguimos analisar das possibilidades de formação de caminhos diferenciais válidos, parece improvável que exista um caminho diferencial válido com menos do que 38 portas AND ativas em 16 rodadas quando utilizada a nova tabela de deslocamento de bits.

A tabela 5 mostra, para cada valor de comprimento do resumo da mensagem d, quantas rodadas seriam necessárias para garantir a segurança da função de compressão do MD6 contra um ataque diferencial padrão, considerando a possibilidade de que o número mínimo de portas AND ativas em 16 rodadas com a nova tabela de deslocamento de bits seja 38, e compara essa quantidade de rodadas com a quantidade de rodadas necessárias quando a tabela de deslocamento de bits original do MD6 é utilizada.

Tabela 5. Número mínimo de rodadas r para cada valor de d quando a tabela original do MD6 é utilizada e portanto existe um caminho diferencial com 26 portas AND ativas em 16 rodadas, comparado ao número mínimo de rodadas considerando a possibilidade de que o número mínimo de portas AND ativas em 16 rodadas seja 38 quando utilizada a nova tabela de deslocamento de bits (número mínimo de rodadas já somado à margem de segurança de 15 rodadas).

d	min AAGs	r min original	r min com nova tabela	redução
40	20	29	29	0%
80	40	43	37	14%
128	64	57	46	19%
160	80	66	55	17%
224	112	87	63	28%
256	128	90	76	16%
384	192	132	101	23%
512	256	165	127	23%

Segue um exemplo de como foram calculados os números de rodadas na tabela 5: precisamos de 5 conjuntos de 16 rodadas mais 1 conjunto de 6 rodadas para garantir que haverá no mínimo mínimo 192 portas AND ativas para quando o comprimento do resumo da mensagem é de 384 bits, pois se cada conjunto de 16 rodadas tem no mínimo 38 portas AND ativas e um conjunto de 6 rodadas tem no mínomo 3 portas AND ativas (tabela 2), então em 5 conjuntos de 16 rodadas mais 1 conjunto de 6 rodadas teremos no mínimo $5 \times 38 + 3 = 193$ portas AND ativas. Então, o número mínimo de rodadas deverá ser $5 \times 16 + 6 = 86$. Somando as 15 rodadas da margem de segurança, chegamos em 101 rodadas. As 132 rodadas necessárias quando a tabela de deslocamento de bits original do MD6 é utilizada foi calculada da mesma forma, mas considerando que nesse caso o número mínimo de portas AND ativas em 16 rodadas é 26, e não 38.

4. Conclusão

A eficiência do MD6 é excelente em sistemas com múltiplas unidades de processamento, mas nas plataformas de referência da competição do NIST para escolha do SHA-3 ela não é suficiente para torná-la competitiva.

Ao anunciar que o MD6 não atenderia aos requisitos estabelecidos pelo NIST para a competição de escolha do SHA-3, Ron Rivest alertou que seria extremamente importante que o algoritmo de SHA-3 que viesse a ser escolhido fosse demonstravelmente resistente a ataques diferenciais.

O número de rodadas da função de compressão do MD6 tem um impacto direto na velocidade de processamento, e precisa ser relativamente alto para a demonstração da

resistência do MD6 a ataques diferenciais. A demonstração da resistência a ataques diferenciais exige a comprovação de que em um determinado número de rodadas haverá um número mínimo de portas AND ativas. Seguindo inicialmente as sugestões apresentadas em [Rivest et al. 2008] na página 111, mostramos nesse trabalho que é possível demonstrar a resistência da função de compressão a ataques diferenciais com um número menor de rodadas, mostrando que o número mínimo de portas AND ativas em 16 rodadas pode ser maior se utilizada na função de compressão uma nova tabela de deslocamento de bits (4).

Verificamos se existiam caminhos diferenciais válidos correspondentes aos padrões de peso de caminho diferencial encontrados na busca do limite inferior de portas AND ativas em até 15 rodadas. Constatamos que esses caminhos diferenciais válidos existiam para alguns padrões de peso de caminho diferencial, mas, analisando esses caminhos diferenciais descobrimos que alguns deles só existiam devido a determinadas características da tabela de deslocamento de bits original do MD6 (1), o que identificamos ser uma falha na escolha dos valores da tabela original.

Buscamos por uma nova tabela de deslocamento de bits e encontramos a tabela (4), que não possui as falhas identificadas na tabela original e nem outras possíveis falhas encontradas em outras tabelas durante o processo de busca, e ainda é melhor para a taxa de difusão de bits, que foi o critério usado para a escolha da tabela original.

O uso desta nova tabela não aumenta o número mínimo de portas AND ativas em até 15 rodadas, que foi o número de rodadas analisado originalmente na demonstração da resistência do MD6 a ataques diferencias, mas a nova tabela faz diferença quando o cálculo é feito para 16 rodadas. Quando a tabela original do MD6 é usada, existe um caminho diferencial válido com 26 portas AND ativas em 16 rodadas. Com a nova tabela só conseguimos encontrar um caminho diferencial válido em 16 rodadas com 38 portas AND ativas, e já comprovamos que não existem caminhos válidos com até 27 portas AND ativas. Se confirmado que 38 é o número mínimo de portas AND ativas em 16 rodadas, a nova tabela de deslocamento de bits torna possível a demonstração da resistência do MD6 a ataques diferenciais com um número de rodadas reduzido de acordo com os resultados apresentados na tabela 5.

Referências

Rivest, R., Agre, B., Bailey, D., Crutchfield, C., Dodis, Y., Fleming, K., Khan, A., Krishnamurthy, J., Lin, Y., Reyzin, L., et al. (2008). The MD6 hash function A proposal to NIST for SHA-3. *Submission to NIST*.