Problem 11. «Unbalanced compression»

Michelle developed a new cryptographic hash function based on the Merkle-Damgård construction. A message is split into 224-bit blocks, while a padding scheme is applied to make sure the splitting occurs regardless the message's length. A compression function takes a message block as an input and results in a 128-bit output.

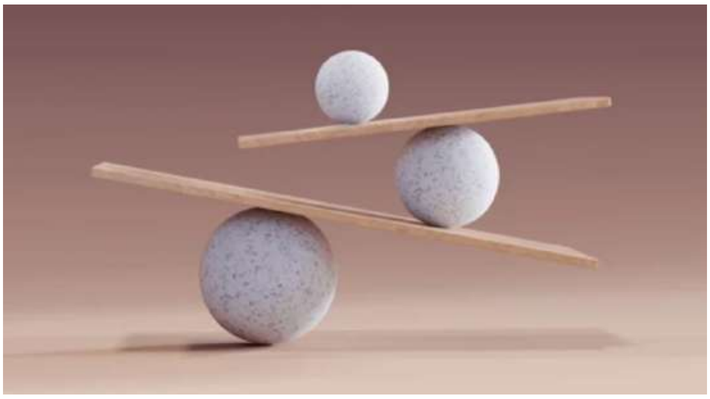
Algorithm 1 (see below) presents the compression function when it processes the first message block M. The block is divided into seven 32-bit integers $M_0, \ldots, M_6$. On the first message block, 32-bit integers A, B, C, D are initialized with constants and then during 40 rounds they are updated. Finally, the output is formed as a concatenation of A, B, C, D. The function $[txyz\,e\;s]_F$ stands for $t = (t + F(x, y, z) + e) \lll s$, where "$\lll s$" is the circular shifting to the left by s bits position, while "+" is the addition modulo 2^{32}. The functions $[txyz\;e\;s]_G$ and $[txyz\;e\;s]_H$ stand for $t = (t+G(x, y, z)+e+\texttt{0x5a827999}) \lll s$ and $t = (t + H(x, y, z) + e + \texttt{0x6ed9eba1}) \lll s$, respectively.

Also let $F(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$; $G(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$; $H(x, y, z) = x \oplus y \oplus z$.

In the pseudocode, "$\leftarrow$" stands for assignment, while "=" stands for equality.

For example, "[DABC M_4 7]$_F$; $D = K$" means that D is updated by $D \leftarrow (D+F(A, B, C)+M_4) \lll 7$ and then it turns out that the updated D is equal to K, so $K = (D + F(A, B, C) + M_4) \lll 7$.

Michelle managed to find all preimages for several compression function's outputs using algebraic cryptanalysis. It turned out there are usually 0, 1, or 2 preimages in total for an output, i.e. there are outputs with no preimages. However, Michelle expected about 2^{96} preimages for each output because the compression function maps 2^{224} onto 2^{128}. Please, help Michelle to find out why the compression function is unbalanced.



Algorithm 1 A compression function on the first 224-bit message block.

Input: 224-bit message block M.
Output: 128-bit output out.

$AA \leftarrow A \leftarrow$ `0x67452301`; $BB \leftarrow B \leftarrow$ `0xefcdab89`
$CC \leftarrow C \leftarrow$ `0x98badcfe`; $DD \leftarrow D \leftarrow$ `0x10325476`
$K \leftarrow$ `0xffffffff`; $P \leftarrow$ `0xa57d8668`
$[\text{ABCD P } 3]_F$ $[\text{DABC P } 7]_F$ $[\text{CDAB P } 11]_F$ $[\text{BCDA } M_0 \ 19]_F$ ▷ Rounds 1-4
$[\text{ABCD P } 3]_F$ $[\text{DABC P } 7]_F$ $[\text{CDAB P } 11]_F$ $[\text{BCDA } M_1 \ 19]_F$
$[\text{ABCD P } 3]_F$ $[\text{DABC P } 7]_F$ $[\text{CDAB P } 11]_F$ $[\text{BCDA } M_2 \ 19]_F$
$[\text{ABCD } M_3 \ 3]_F$; $A = K$ ▷ The updated A equals K
$[\text{DABC } M_4 \ 7]_F$; $D = K$ ▷ The updated D equals K
$[\text{CDAB } M_5 \ 11]_F$; $C = K$ ▷ The updated C equals K
$[\text{BCDA } M_6 \ 19]_F$
$[\text{ABCD P } 3]_G$; $A = K$ ▷ The updated A equals K
$[\text{DABC P } 5]_G$; $D = K$ ▷ The updated D equals K
$[\text{CDAB P } 9]_G$; $C = K$ ▷ The updated C equals K
$[\text{BCDA } M_3 \ 13]_G$
$[\text{ABCD P } 3]_G$; $A = K$ ▷ The updated A equals K
$[\text{DABC P } 5]_G$; $D = K$ ▷ The updated D equals K
$[\text{CDAB P } 9]_G$; $C = K$ ▷ The updated C equals K
$[\text{BCDA } M_4 \ 13]_G$
$[\text{ABCD P } 3]_G$; $A = K$ ▷ The updated A equals K
$[\text{DABC P } 5]_G$; $D = K$ ▷ The updated D equals K
$[\text{CDAB P } 9]_G$; $C = K$ ▷ The updated C equals K
$[\text{BCDA } M_5 \ 13]_G$ ▷ Round 28
$[\text{ABCD } M_0 \ 3]_G$ $[\text{DABC } M_1 \ 5]_G$ $[\text{CDAB } M_2 \ 9]_G$ $[\text{BCDA } M_6 \ 13]_G$
$[\text{ABCD P } 3]_H$ $[\text{DABC P } 9]_H$ $[\text{CDAB P } 11]_H$ $[\text{BCDA } M_3 \ 15]_H$
$[\text{ABCD P } 3]_H$ $[\text{DABC P } 9]_H$ $[\text{CDAB P } 11]_H$ $[\text{BCDA } M_5 \ 15]_H$
$A \leftarrow A + AA$; $B \leftarrow B + BB$
$C \leftarrow C + CC$; $D \leftarrow D + DD$
$out \leftarrow concatenation(A, B, C, D)$