F.F.T. Hashing is not Collision-free

T. BARITAUD * , H. GILBERT * , M. GIRAULT **

(*) CNET PAA / TSA / SRC
38 - 40, avenue du Général Leclerc
92131 ISSY LES MOULINEAUX (France)

(**) SEPT PEM
42, rue des Coutures
BP 6243
14066 CAEN (France)

Abstract

The FFT Hashing Function proposed by C.P. Schnorr [1] hashes messages of arbitrary length into a 128-bit hash value. In this paper, we show that this function is not collision free, and we give an example of two distinct 256-bit messages with the same hash value. Finding a collision (in fact a large family of colliding messages) requires approximately 2^{23} *partial computations of the hash function, and takes a few hours on a SUN3- workstation, and less than an hour on a SPARC-workstation.*

A similar result discovered independently has been announced at the Asiacrypt'91 rump session by Daemen-Bosselaers-Govaerts-Vandewalle [2].

1 The FFT Hashing Function

1.1 The Hash algorithm

Let the message be given as a bit string $m_1 m_2 ... m_t$ of t bit.

The message is first padded so that its length (in bits) becomes a multiple of 128. Let the padded message $M_1 M_2 ... M_n$ consist of n blocks $M_1, ..., M_n$, each of the M_i (i=1, ... ,n) being 128-bit long.

The algorithm uses a constant initial value H_0 given in hexadecimal as

$$H_0 = 0123\ 4567\ 89ab\ cdef\ fedc\ ba98\ 7654\ 3210 \ \text{in}\ \{0,1\}^{128}.$$

Let p be the prime $65537 = 2^{16} + 1$.

We will use the Fourier transform $FT_8 : \{0, \dots, p-1\}^8 \text{------>} \{0, \dots, p-1\}^8$

$$(a_0, \dots, a_7) \text{------>} (b_0, \dots, b_7)$$

with $b_i = \sum_{j=0}^{7} 2^{4ij} a_j \mod p$, for $i = 0, \dots, 7$.

Algorithm for the hash function h :

INPUT : $M_1 M_2 \dots M_n$ in $\{0,1\}^{n.128}$ (a padded message)

DO : $H_i = g(H_{i-1}, M_i)$ for $i = 1, \dots, n$

OUTPUT : $h(M) := H_n$

Algorithm for $g : Z_p^{16} \text{------>} \{0,1\}^{8.16}$

INPUT $(c_0, \dots, c_{15})$ in $\{0,1\}^{16.16}$

1. $(c_0, c_2, \dots, c_{14}) := FT_8(c_0, c_2, \dots, c_{14})$
2. FOR $i = 0, \dots, 15$ DO

   $c_i := c_i + c_{i-1}c_{i-2} + c_{c_{i-3}} + 2^i \pmod p$

   (The lower indices $i, i-1, i-2, i-3, c_{i-3}$ are taken modulo 16)
3. REPEAT steps 1 and 2

OUTPUT $\overline{c}_i := c_i \mod 2^{16}$, for $i = 8, \dots, 15$ (an element of $\{0,1\}^{8.16}$)

1.2 Notations

For a better clarity of our explanation, we will denote by c_i^0 (i=0, ... ,15) the initial c_i values, and we will denote by step 3 (resp. step 4) the second pass of step 1 (resp. step2) in the algorithm for g.

When it will be necessary to avoid any kind of slip, we will denote by c_i^k (i=0, ... ,15 ; k=0, ... ,4) the c_i intermediate value, after step k.

In order to simplify the expressions, we are using the following notations :

- The additions (x+y), multiplications (x.y) and exponentiations (x^y) are implicitly made modulo p, except when the operands are lower indices.
- The = symbol denotes that the right and the left terms are congruent modulo p.
- For lower indices the additions (i+j) and substractions (i-j) are implicitly made modulo 16, and the ≡ symbol denotes that the right and the left terms are congruent modulo 16.

1.3 Preliminary remarks

The difficulty of finding collisions is related to the diffusion properties of the hashing function, i.e. the influence of a modification of an intermediate variable on the subsequent variables of the calculation.

Remark 1 (limitation on the diffusion at steps 1 and 3)

At step 1 and 3, the input values $c_1, c_2, \ldots, c_{15}$ are kept unchanged.

Remark 2 (limitation on the diffusion at steps 2 and 4)

The diffusion introduced by the $c_{i-1}c_{i-2}$ terms in the recurrence for steps 2 and 4 can sometimes be cancelled (if one of values e_{i-1} and c_{i-2} is 0). More precisely, let $(c_0^1, c_1^1, \ldots, e_{15}^1)$ be the input to step 2 :

Proposition 1 : If for a given value i in {1, ... ,14} we have $c_{i-1}^2 = c_{i+1}^2 = 0$ and if $c_{13}^1 \not\equiv i$; $c_{14}^1 \not\equiv i$; $e_{15}^1 \not\equiv i$; $c_j^2 \not\equiv i$ for j in {0, ... ,12}, then the impact of replacing the input value c_i^1 by a new value $e_i^1 + \Delta c_i^1$ such that $c_i^1 + \Delta c_i^1 \equiv c_i^1$, is limited to the output value c_i^2 (that means c_j^2 are not modified for j ≠i).

Proposition 2 : If $e_{14}^1 = c_0^2 = 0$ and if $c_j^2 \not\equiv 15$ for j in {1, ... ,11} then the impact of replacing the input value e_{15}^1 by a new value $c_{15}^1 + \Delta c_{15}^1$ such that $c_{15}^1 + \Delta c_{15}^1 \equiv c_{15}^1$, is limited to the output value c_{15}^2.

Similarly, let $(c_1^3, c_2^3, \ldots, c_{15}^3)$ be the input to step 4 :

Proposition 1' : If for a given value i in {1, ... ,14} we have $e_{i-1}^4 = e_{i+1}^4 = 0$ and if $e_{13}^3 \not\equiv i$; $c_{14}^3 \not\equiv i$; $c_{15}^3 \not\equiv i$; $c_j^4 \not\equiv i$ for j in {0, ... ,12}, then the impact of replacing the input value c_i^3 by a new value $e_i^3 + \Delta c_i^3$ such that $c_i^3 + \Delta c_i^3 \equiv e_i^3$, is limited to the output value c_i^4.

Proposition 2' : If $e^3_{14} = e^4_0 = 0$ and if $e^4_j \not\equiv 15$ for j in {1, ... ,11} then the impact of replacing the input value e^3_{15} by a new value $e^3_{15} + \Delta e^3_{15}$ such that $e^3_{15} + \Delta e^3_{15} \equiv e^3_{15}$ is limited to the output value e^4_{15}.

2 Construction of two colliding messages

2.1 Construction of a partial collision

We first find two 128-bit blocks M_1 and M'_1 which hash values $H_1 = (\overline{e}^4_8, \ldots, \overline{e}^4_{15})$ and $H'_1 = (\overline{e'}^4_8, \ldots, \overline{e'}^4_{15})$ differ only by their right components $\overline{e}^4_{15}$ and $\overline{e'}^4_{15}$. We will later refer to this property in saying that M_1 and M'_1 realize a partial collision.

Our technique for finding M_1 and M'_1 is the following : we search M_1 values such that $e^1_{14} = 0$; $e^2_0 = 0$; $e^3_{14} = 0$; $e^4_0 = 0$. The propositions 2 and 2' suggest that for such a message $M_1 = (e^0_8, \ldots, e^0_{14}, e^0_{15})$, M_1 and the message $M'_1 = (e^0_8, \ldots, e^0_{14}, e^0_{15} + 16)$ realize a partial collision with a significant probability (approximately 1/8).

There are two main steps for finding M_1.

Step1 : Selection of $e^0_8, e^0_{10}, e^0_{12}$ and e^0_{14}

Arbitrary (e.g. random) values are taken for e^0_{12} and e^0_{14}. The values of e^0_8 and e^0_{10} are then deduced from these values by solving the following linear system :

$$\begin{cases} e^1_{14} = 0 & (1) \\ e^1_0 = -1 & (2) \end{cases}$$

Proposition 3 :
If $e^0_{13} \equiv 14$ then $e^1_{14} = 0$ and $e^2_0 = 0$ independently of the values of $e^0_9, e^0_{11}, e^0_{13}, e^0_{15}$.

Proof : This is a direct consequence of the definition of the g function.

<u>Step 2</u> : Selection of $c_9^0, c_{11}^0, c_{13}^0, c_{15}^0$

The values of $c_8^0, c_{10}^0, c_{12}^0, c_{14}^0$ are taken from Step 1 .

We fix the values of $c_{11}^0 = 0$ and $c_{15}^0 = 0$. An arbitrary (e.g random) value is taken for c_9^0. We first calculate the c_{12}^2 and c_{14}^3 values corresponding to the chosen value of c_9^0, c_{11}^0 and c_{15}^0 and to the temporary value $c_{13}^0 = 14$. Based on these preliminary calculations, we "correct" the temporary value $c_{13}^0 = 14$ by a quantity Δc_{13}^0, i.e. we replace the value $c_{13}^0 = 14$ by the value $c_{13}^0 = 14 + \Delta c_{13}^0$, and we leave the other input values unchanged. We denote by Δc_j^i $(0 \leq i \leq 4\ ;\ 0 \leq j \leq 15)$ the corresponding variations of the intermediate variables in the H_1 calculation. We select Δc_{13}^0 in such a way that the quantity $c_{14}^3 + \Delta c_{14}^3$ (i.e the new value of c_{14}^3) is equal to zero with a good probability.

<u>Proposition 4</u> : If $c_{12}^2 \neq 0$ and $\frac{-c_{14}^3}{2^{4.7.7}\, c_{12}^2} \equiv 0$ and $c_j^2 \not\equiv 13$ for $1 \leq j \leq 11$ then the above values of c_{15}^1, c_0^2 and the value $\Delta c_{13}^0 = \frac{-c_{14}^3}{2^{4.7.7}\, c_{12}^2}$ lead to the three relations

$$\begin{cases} c_{14}^1 + \Delta c_{14}^1 = 0 & \text{(a)} \\ c_0^2 + \Delta c_0^2 = 0 & \text{(b)} \\ c_{14}^3 + \Delta c_{14}^3 = 0 & \text{(c)} \end{cases}$$

<u>Proof</u> : (a) is straightforward; (b) and (c) are direct consequences of the following relations, which result from the definition of the g function :

$$\Delta c_{j-2}^2 = 0 \text{ for } 0 \leq j \leq 12 \quad ; \quad \Delta c_{13}^2 = \Delta c_{13}^0 \quad ; \quad \Delta c_{14}^2 = c_{12}^2 \cdot \Delta c_{13}^2 \quad ; \quad \Delta c_{14}^3 = 2^{4.7.7} \cdot \Delta c_{14}^2$$

We performed a large number n_1 of trials of step 1. For each trial of step 1, we made a large number n_2 of trials of step 2. The success probability of step 2, i.e the probability that the trial of a c_9^0 value leads to a message such that (a), (b) and (c) are realized is slightly less than 1/16 (since the strongest condition in proposition 2 is : $\frac{-c_{14}^{3}}{2^{4.4.7} c_{12}^{2}} \equiv 0$). Therefore the probability that a step 2 trial leads to a message M_1 such that $c_{14}^{1} = c_{0}^{2} = c_{14}^{3} = c_{0}^{4} = 0$ is slightly less than $1/16 . 2^{-16} = 2^{-20}$.

Moreover, the probability that such a message M_1 leads to a partial collision is basically the probability that none of the c_{i-3} mod 16 indices occurring in the calculation of c_0^2 to c_{15}^2 and c_0^4 to c_{15}^4 takes the value 15, which is close to 1/8. So, in summary, approximativelv 2^{23} partial computations of the g function were necessary to obtain a suitable message $M_1 = (c_8^0, \ldots, c_{14}^0, c_{15}^0)$, such that M_1 and the message $M'_1 = (c_8^0, \ldots, c_{14}^0, c_{15}^0 + 16)$ lead to partially colliding hash values $H_1 = (\overline{c}_8^{\,4}, \ldots, \overline{c}_{15}^{\,4})$ and $H'_1 = (\overline{c}_8^{\,4}, \ldots, \overline{c}_{15}^{\,4} + 16)$.

2.2 Construction of a full collision using a partial collision

We now show how to find a 128-bit message $M_2 = (c_8^0, \ldots, c_{15}^0)$ such that the previously obtained hash values H_1 and H'_1 (denoted in this section by $(c_0^0, \ldots, c_7^0)$ and $(c'^{\,0}_1, \ldots, c'^{\,0}_6, c'^{\,0}_7) = (c_1^0, \ldots, c_6^0, c_7^0 + 16)$) respectively lead to the same hash value H_2 (when combined with M_2): $g(H_1, M_2) = g(H'_1, M_2)$.

Our technique for finding M_2 is quite similar to the one used for finding M_1 and M'_1. Let us denote by c_j^i (resp $c'^{\,i}_j$) $(0 \le i \le 4, 0 \le j \le 15)$ the intermediate variables of the calculations of $g(H_1, M_2)$ (resp $g(H'_1, M_2)$).

We search M_2 values such that $c_6^2 = c_8^2 = c_6^4 = c_8^4 = 0$. The propositions 1 and 1' suggest that the probability that the 16-uples $(c_0^4, \ldots, c_{15}^4)$ and $(c'^{\,4}_0, \ldots, c'^{\,4}_{15})$ differ only by their components c_7^4 and $c'^{\,4}_7$ which implies that the probability to have $g(H_1, M_2) = g(H'_1, M_2)$ is quite substantial, approximatively 1/8.

There are two main steps for the search of M_2 :

Step 1 : Selection of $c_8^0, c_{10}^0, c_{12}^0, c_{14}^0, c_9^0$.

An arbitrary (e.g random) value is taken for c_{14}^0. The values of $c_8^0, c_{10}^0, c_{12}^0$ are deduced from c_{14}^0 by solving the following linear system :

$$\begin{cases} c_{14}^1 = 0 & (3) \\ c_0^1 = -1 & (4) \\ c_8^1 = -2^8 & (5) \end{cases}$$

A preliminary calculation, where c_9^0, c_{11}^0 and c_{15}^0 are set to the temporary value 0 and c_{13}^0 is set to the temporary value 14, is made. The obtained value of c_6^2, denoted by δ, is kept.

Proposition 5 : If $c_8^0, c_{10}^0, c_{12}^0, c_{14}^0$ are solutions of (3), (4), (5) and if in addition the values $c_9^0 = p-\delta$, $c_{11}^0 = 0$, $c_{13}^0 = 14$, $c_{15}^0 = 0$ lead to intermediate values such that : c_1^2 mod 16 is not in {9,11,13,15}; c_2^2 mod 16 is not in {9,11,13,15}; $c_3^2 \equiv 9$ mod 16; c_4^2 mod 16 is not in {9,11,13,15}; c_5^2 mod 16 is in {0,6,14}, then if we fix the value $c_9^0 = p-\delta$, for any value of $c_{13}^0 \equiv 14$ and for any value of c_{15}^0 such that $c_{15}^0 \equiv 0$ we have :

$$c_{14}^1 = 0 \ ; \qquad c_0^2 = 0 \ ; \qquad c_6^2 = 0 \ ; \qquad c_8^2 = 0 \ .$$

Proof : The proof of this proposition is easy. Finding the $c_8^0, c_{10}^0, c_{12}^0, c_{14}^0$ and c_9^0 values satisfying the conditions of the above proposition is quite easy, and requires the trial of a few hundreds c_{14}^0 values.

Step 2 : Selection of $c_{11}^0, c_{13}^0, c_{15}^0$

The values of $c_8^0, c_{10}^0, c_{12}^0, c_{14}^0, c_9^0$ are taken from Step 1 ; these values are assumed to realize the conditions of the above proposition.

An arbitrary (e.g random) value is taken for c_{11}^0. A preliminary calculation is made, using the selected c_{11}^0 value and the temporary values $c_{13}^0 = 14$; $c_{15}^0 = 0$. The corresponding values of c_{12}^2 and c_8^3 are kept.

Based on these preliminary calculations, we "correct" the temporary value of c_{13}^0 by a quantity Δc_{13}^0 and we also consider new values $c_{15}^0+\Delta c_{15}^0$ for c_{15}^0. The variation Δc_{13}^0 is selected in such a way that for any Δc_{15}^0 value satisfying $\Delta c_{15}^0 \equiv 0$, the new value $e_8^3+\Delta e_8^3$ of e_8^3 is equal to -2^8 with a substantial probability.

Proposition 6 : If $e_{12}^2 \neq 0$ and $\dfrac{-2^8 - e_8^3}{2^{4.4.7}\, e_{12}^2} \equiv 0$ and e_j^2 mod 16 is not in {13,15} for $1 \leq j \leq 11$ then for any variation $\Delta c_{15}^0 \equiv 0$ on c_{15}^0 such that $c_{15}^2 + \Delta c_{15}^0 < p$ and $c_{15}^4 + \Delta c_{15}^0 < p$, the variation

$$\Delta e_{13}^0 = \frac{-2^8 - e_8^3}{2^{4.4.7}\, e_{12}^2}$$

on the c_{13}^0 value leads to the following new values :

$$e_{14}^1 + \Delta c_{14}^1 = 0 \;;\quad c_0^2 + \Delta e_0^2 = 0 \;;\quad c_6^2 + \Delta c_6^2 = 0 \;;\quad c_8^2 + \Delta c_8^2 = 0 \;;\quad c_8^3 + \Delta c_8^3 = -2^8 .$$

We performed a number n_1 of trials of step 1. For each successful trial of step 1, we made a large number n_2 of trials of c_{11}^0 values at step 2. For those c_{11}^0 values satisfying the conditions of the above proposition, we made a large number n_3 of trials of new c_{15}^0 values such that $\Delta c_{15}^0 \equiv 0$. The probability that the trial of a new Δc_{15}^0 value leads to intermediate variables satisfying the four equations $c_6^2=0$; $c_8^2=0$; $c_6^4=0$; $e_8^4=0$ is basically the probability that randomly tried c_6^4 and c_5^4 values satisfy $c_6^4 = 0$ and $c_5^4 \equiv 6$; the order of magnitude of this probability is therefore 2^{-20}.

Moreover, the probability that a message M_2 satisfying the four equations $c_6^2=0$; $c_8^2=0$; $c_6^4=0$; $c_8^4=0$ leads to a full collision $g(H_1,M_2) = g(H_1,M_2')$ is basically the probability that none of the c_{i-3} mod 16 indices occurring in the calculation of c_0^2 to e_{15}^2 and of c_0^4 to c_{15}^4 takes the value 15, which is close to 1/8. So in summary approximatively 2^{23} partial computations of the g function are necessary to obtain a message M_2 giving a full collision.

2.3 Implementation details

The above attack method was implemented using a non-optimized Pascal program. The search for a collision took a few hours on a SUN3 workstation and less than an hour on a SPARC workstation. We provide in annex the detail of the intermediate calculations for two colliding messages M_1M_2 and M'_1M_2, of two 128-bit blocks each.

Note that for many other values M''_1 of the form $(e_0^0, \ldots, e_{15}^0 + k.16)$ (k : an integer) of the first 128-bit block, the message M''_1M_2 leads to the same hash value as M_1M_2 : the observed phenomenon is in fact a multiple collision.

3 Conclusions

The attack described in this paper takes advantage of the two following weaknesses of the FFT-Hashing algorithm :

- the influence of the term $e_{e_{i-3}}$ in the recurrence $e_i := e_i + e_{i-1}e_{i-2} + e_{e_{i-3}} + 2^i$ (mod p) on the security of the algorithm is rather negative (see for example the method to obtain $e_6^2 = 0$ (or $e_8^2 = 0$) at step 1 of Section 2.2).
- as mentioned in Section 1.3, the diffusion introduced by the four steps of the algorithm is quite limited. In particular, the FT_8 Fourier transform acts only on half of the intermediate values $(e_0, \ldots, e_{15})$, namely the 8 values $e_0, e_2, \ldots, e_{14}$.

This suggests that quite simple modifications might result in a substantial improvement of the security of the FFT-Hashing algorithm.

4 Acknowledgements

The autors are greateful to Jacques BURGER (SEPT PEM, 42 rue des Coutures, BP 6243, 14066 CAEN, France) for the Sparc implementation as well as useful discussions.

5 References

[1] : C.P. SCHNORR; FFT-Hashing : An Efficient Cryptographic Hash Function; July 15, 1991 (This paper was presented at the rump session of the CRYPTO'91 Conference, Santa Barbara, August, 11-15, 1991)

[2] : DAEMEN - BOSSELAERS - GOVAERTS - VANDEWALLE : Announcement made at the rump session of the ASIACRYPT '91 Conference, Fujiyoshida, Japan, November 11-14, 1991)

ANNEX

```
FIRST MESSAGE  M = M1 M2      with

        M1 =   F95A  807A   26A     0   440  365E     0     0

        M2 =   1537  5202  3284   358  5D1C  959E  6D6B  75E0

calculation of H1 :
-----------------

        H0 =    123  4567  89AB  CDEF  FEDC  BA98  7654  3210

        M1 =   F95A  807A   26A     0   440  365E     0     0

    step 1:   10000  4567  4F72  CDEF  B84C  BA98  D98A  3210
               FB30  807A  F62E     0  3677  365E     0     0

    step 2:       0  4569  4F76  1DD1  6CEA  F49C  1DB9  7D13
               ADDC   156  5AFE  CD52  A692  158A  4626  B80B

    step 1:    CFA9  4569  2466  1DD1  2F1A  F49C  F3D7  7D13
               B305   156  3057  CD52   5A7  158A     0  B80B

    step 2:       0  456B  F1BC  91E1  64F8  F6D2  FB99  A787
               7DCA  CDE2  4508  3BE5  8F64  E23C  9B8A  5B06


        H1 =   7DCA  CDE2  4508  3BE5  8F64  E23C  9B8A  5BE6

calculation of H2 :
-----------------

        H1 =   7DCA  CDE2  4508  3BE5  8F64  E23C  9B8A  5BE6

        M2 =   1537  5202  3284   358  5D1C  959E  6D6B  75E0

    step 1:   10000  CDE2  C5BE  3BE5  3E13  E23C  418A  5BE6
               FF01  5202  9B04   358   EF0  959E     0  75E0

    step 2:       0  CDE4  C5C2  17A9  6501  6370     0  2A49
                  0  5402  F306  99A5  8BB5  9A6E  38EF  73A9

    step 1:    E26B  CDE4  8B79  17A9  E6CC  6370  E7C2  2A49
               FF01  5402   CD5  99A5  37CB  9A6E  7FF2  73A9

    step 2:    5551  E84C  4E20  EA99  C82F  9B86     0  9E72
                  0  AB53  5EF5  27D8  9554   995  983F  89CF


        H2 =      0  AB53  5EF5  27D8  9554   995  983F  89CF

HASHED MESSAGE : 0  AB53  5EF5  27D8  9554   995  983F  89CF
```

```
SECOND MESSAGE  M = M1 M2    with

        M1 =   F95A  807A   26A     0   440  365E     0    10

        M2 =   1537  5202  3284   358  5D1C  959E  6D6B  75E0

calculation of H1 :
-----------------

        H0 =    123  4567  89AB  CDEF  FEDC  BA98  7654  3210

        M1 =   F95A  807A   26A     0   440  365E     0    10

    step 1:   10000  4567  4F72  CDEF  B84C  BA98  D98A  3210
               FB30  807A  F62E     0  3677  365E     0    10

    step 2:       0  4569  4F76  1DD1  6CEA  F49C  1DB9  7D13
               ADDC   156  5AFE  CD52  A692  158A  4626  B81B

    step 1:    CFA9  4569  2466  1DD1  2F1A  F49C  F3D7  7D13
               B305   156  3057  CD52   5A7  158A     0  B81B

    step 2:       0  456B  F1BC  91E1  64F8  F6D2  FB99  A787
               7DCA  CDE2  4508  3BE5  8F64  E23C  9B8A  5BF6


        H1 =   7DCA  CDE2  4508  3BE5  8F64  E23C  9B8A  5BF6

calculation of H2 :
-----------------

        H1 =   7DCA  CDE2  4508  3BE5  8F64  E23C  9B8A  5BF6

        M2 =   1537  5202  3284   358  5D1C  959E  6D6B  75E0

    step 1:   10000  CDE2  C5BE  3BE5  3E13  E23C  418A  5BF6
               FF01  5202  9B04   358   EF0  959E     0  75E0

    step 2:       0  CDE4  C5C2  17A9  6501  6370     0  2A59
                  0  5402  F306  99A5  8BB5  9A6E  38EF  73A9

    step 1:    E26B  CDE4  8B79  17A9  E6CC  6370  E7C2  2A59
               FF01  5402   CD5  99A5  37CB  9A6E  7FF2  73A9

    step 2:    5551  E84C  4E20  EA99  C82F  9B86     0  9E82
                  0  AB53  5EF5  27D8  9554   995  983F  89CF


        H2 =      0  AB53  5EF5  27D8  9554   995  983F  89CF

HASHED MESSAGE :  0  AB53  5EF5  27D8  9554   995  983F  89CF
```