FFT-Hash-II is not yet Collision-free

S. VAUDENAY

Laboratoire d'Informatique, URA 1327 du CNRS
Département de Mathématiques et d'Informatique
Ecole Normale Supérieure

LIENS - 92 - 17

September 1992

FFT-Hash-II is not yet Collision-free

Serge Vaudenay

LIENS *, 45 rue d'Ulm, 75230 Paris cedex 05, France

Abstract. In this paper, we show that the FFT-Hash function proposed by Schnorr [2] is not collision free. Finding a collision requires about 2^{24} computation of the basic function of FFT. This can be done in few hours on a SUN4-workstation. In fact, it is at most as strong as a one-way hash function which returns a 48 bits length value. Thus, we can invert the proposed FFT hash-function with 2^{48} basic computations. Some simple improvements of the FFT hash function are also proposed to try to get rid of the weaknesses of FFT.

History

The first version of FFT-Hashing was proposed by Schnorr during the rump session of Crypto'91 [1]. This function has been shown not to be collision free at Eurocrypt'92 [3]. An improvement of the function has been proposed the same day [2] without the weaknesses discovered. However, FFT-Hashing has still some other weaknesses as it is proved in this paper.

1 FFT-Hash-II, Notations

The FFT-hash function is built on a basic function $< . >$ which takes one 128-bits long hash block H and one 128-bits long message block M, and return a 128-bits long hash block $< H, M >$. The hash value of n message blocks $M_1, \ldots, M_n$ is $< \ldots << H_0, M_1 >, M_2 >, \ldots, M_n >$ where H_0 is a constant given in hexadecimal by :

$$H_0 = 0123\ 4567\ 89ab\ cdef\ fedc\ ba98\ 7654\ 3210$$

The basic function is defined by two one-to-one functions Rec and FT2 on the set $(\mathrm{GF}_p)^{16}$ where $p = 2^{16} + 1$. The concatenation HM defines 16 16-bits numbers which represents 16 numbers in GF_p between 0 and $p - 2$. $(\mathrm{Rec} \circ \mathrm{FT2} \circ \mathrm{Rec})(HM)$ defines 16 numbers of GF_p. The last 8 numbers taken modulo 2^{16} are the result $< H, M >$.

* The *Laboratoire d'Informatique de l'Ecole Normale Supérieure* is a research group affiliated with the CNRS

We define the following notations :

$$\begin{aligned} A(M) &= H_0 M \\ B(M) &= \mathrm{Rec}(A(M)) \\ C(M) &= \mathrm{FT2}(B(M)) \\ D(M) &= \mathrm{Rec}(C(M)) \end{aligned}$$

So, $< H_0, M >$ is the last 8 numbers of $D(M)$ taken modulo 2^{16}. We define X_i the i-th number of X (from 0 to 15), and $X[i,j]$ the list of the i-th to the j-th number of X.

If $x_i \in \mathrm{GF}_p$, $i = 0, \ldots, 15$, we define $y_{-3} = x_{13}$, $y_{-2} = x_{14}$, and $y_{-1} = x_{15}$. Then, following Schnorr :

$$y_i = x_i + y_{i-1}^* y_{i-2}^* + y_{i-3} + 2^i \tag{1}$$

where $y^* = 1$ if $y = 0$ and $y^* = y$ otherwise. Then, we let :

$$\mathrm{Rec}(x_0, \ldots, x_{15}) = y_0, \ldots, y_{15}$$

If $x_i \in \mathrm{GF}_p$, $i = 0, \ldots, 7$, we define :

$$y_j = \sum_{i=0}^{7} \omega^{ij} x_i$$

where $\omega = 2^4$. Then, we define $FT(x_0, \ldots, x_7) = y_0, \ldots, y_7$.

If $x_i \in \mathrm{GF}_p$, $i = 0, \ldots, 15$, we define $y_0, y_2, \ldots, y_{14} = FT(x_0, x_2, \ldots, x_{14})$ and $y_1, y_3, \ldots, y_{15} = FT(x_1, x_3, \ldots, x_{15})$. Then, we define $\mathrm{FT2}(x_0, \ldots, x_{15}) = y_0, \ldots, y_{15}$.

2 Basic Remarks

If we want to find a collision to the hash function, we may look for a pair (x, x') of two 128-bits strings such that $< H_0, x > = < H_0, x' >$. In fact, we will look for x and x' such that $D(x)[8,15] = D(x')[8,15]$.

First, we notice that we have necessarily $C(x)[11,15] = C(x')[11,15]$. In one direction, we show that $C(x)_i = C(x')_i$ for $i = 11, \ldots, 15$. This is due to the equation :

$$C_i = D_i - D_{i-1}^* D_{i-2}^* - D_{i-3} - 2^i$$

Conversely, if we have both $C(x)[11,15] = C(x')[11,15]$ and $D(x)[8,10] = D(x')[8,10]$, then we have $D(x)[8,15] = D(x')[8,15]$.

Moreover, we notice on the equation 1 that $B(x)[0,7]$ is a function of $x[5,7]$ only. Let us denote :

$$B(x)[0,7] = g(x[5,7])$$

Finally, we notice that $FT2$ is a linear function.

3 Breaking *FFT*

3.1 Outlines

If we get a set of 3.2^{24} strings x such that $C(x)[11,15]$ is a particular string R chosen arbitrarily[2], we will have a collision on $D(x)[8,10]$ with probability 99% thanks to the birthday paradox. We will describe an algorithm which gives some x with the definitively chosen R for any $x[5,7] = abc$.

Given $abc = x[5,7]$, we can compute $B(x)[0,7] = g(abc)$. If we denote $y = B(x)[8,15]$, the following equation is a linear equation in y ;

$$FT2(g(abc)y)[11,15] = R \tag{2}$$

We can define a function ϕ_R and three vectors U_e, U_o, U'_e such that :

$$(2) \iff \exists \lambda, \lambda', \mu \quad y = \phi_R(abc) + \lambda U_e + \lambda' U'_e + \mu U_o$$

(see section 3.2).

Finally, the system :

$$\left\{ \begin{array}{r} x[5,7] = abc \\ C(x)[11,15] = R \end{array} \right.$$

is equivalent to the system :

$$\left\{ \begin{array}{l} x[5,7] = abc \\ \qquad y = \phi_R(abc) + \lambda U_e + \lambda' U'_e + \mu U_o \\ \quad H_0 x = \mathrm{Rec}^{-1}(g(abc)y) \end{array} \right.$$

Which is equivalent to :

$$\left\{ \begin{array}{l} \qquad y = \phi_R(abc) + \lambda U_e + \lambda' U'_e + \mu U_o \\ \quad y_{13} = a + y^*_{12} y^*_{11} + y_{10} + 2^{13} \\ \quad y_{14} = b + y^*_{13} y^*_{12} + y_{11} + 2^{14} \\ \quad y_{15} = c + y^*_{14} y^*_{13} + y_{12} + 2^{15} \\ x[5,7] = abc \\ x[0,4] = \mathrm{Rec}^{-1}(g(abc)y)[8,12] \end{array} \right. \tag{3}$$

Is we substitute y by the expression of the first equation in the other equations, we obtain a system of three equations of three unknown λ, λ', μ. This system can be shown linear in λ and λ' by a good choice of U_e, U_o and U'_e. Then, this system can have some solutions only if the determinant, which is a degree 2 polynomial in μ is 0. This can gives some μ. Then, the number of (λ, λ') is almost always unique. For more details, see section 3.3.

Finally, this gives 0 or 2 solutions x, with an average number of 1 for a given abc. If we try $1 \leq a < p$, $1 \leq b \leq 768$ and $c = 2$, we have 3.2^{24} abc.

[2] For the collisions found in this paper, R is the image of my phone number by $FT2$.

3.2 Solving (2)

The function $X \longmapsto FT2(X)[11,15]$ is linear, and has a kernel of dimension 3. If we define :

$$U = (0,0,0,0,4081,256,1,61681)$$
$$U' = (0,0,0,0,65521,4352,1,0)$$

we notice that :

$$FT(U) = (482,56863,8160,57887,7682,0,0,0)$$
$$FT(U') = (4337,61202,65503,544,61170,3855,0,0)$$

Let us introduce the following notation :

$$(x_0,\ldots,x_7) \times (y_0,\ldots,y_7) = (x_0,y_0,\ldots,x_7,y_7)$$

We have $FT2(X \times Y) = FT(X) \times FT(Y)$. Thus, we can can define :

$$U_e = U \times 0$$
$$U_o = 0 \times U$$
$$U'_e = U' \times 0$$

So, we have :

$$U_e = (0,0,0,0,0,0,0,0,4081,0,256,0,1,0,61681,0)$$
$$U_o = (0,0,0,0,0,0,0,0,0,4081,0,256,0,1,0,61681)$$
$$U'_e = (0,0,0,0,0,0,0,0,65521,0,4352,0,1,0,0,0)$$

These vectors are a base of the kernel of $X \longmapsto FT2(X)[11,15]$.

If M denotes the matrix of FT, we can write it using four 4×4 blocks :

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$$

If x and y are two vectors of 4 elements, we have :

$$FT(xy)[4,7] = 0 \Longleftrightarrow y = -M_{22}^{-1} M_{21} x$$

Let us define :

$$N = -M_{22}^{-1} M_{21} = \begin{pmatrix} 65281 & 4335 & 289 & 61170 \\ 3823 & 8992 & 53012 & 65248 \\ 8447 & 61748 & 56545 & 4335 \\ 4369 & 57090 & 3823 & 256 \end{pmatrix}$$

Now, if x and y are two vectors of 8 elements, we have :

$$FT2(xy)[8,15] = 0 \Longleftrightarrow y = Nx^0 \times Nx^1$$

Where $x = x^0 \times x^1$. Let us define :

$$\phi_R(abc) = 0(Nx^0 \times Nx^1 + y^0)$$

where $g(abc) = x^0 \times x^1$ and $R = FT2(0y^0)[11,15]$ for an arbitrary y^0 (one's phone number for instance). Then, $\phi_R(abc)$ is a vector which begins by $g(abc)$, and such that $FT2(\phi_R(abc))$ ends by a constant vector R.

So, we have :

$$(2) \Longleftrightarrow \exists \lambda, \lambda', \mu \quad y = \phi_R(abc) + \lambda U_e + \lambda' U'_e + \mu U_o$$

3.3 Solving (3)

If we hope that no y_i $(i = 11, 12, 13, 14)$ is equal to 0 (we may ultimately test this condition, and forget the solutions y which do not pass this test, but this will be very rare), the system :

$$\begin{cases} \quad y = \phi_R(abc) + \lambda U_e + \lambda' U'_e + \mu U_o \\ y_{13} = a + y^*_{12} y^*_{11} + y_{10} + 2^{13} \\ y_{14} = b + y^*_{13} y^*_{12} + y_{11} + 2^{14} \\ y_{15} = c + y^*_{14} y^*_{13} + y_{12} + 2^{15} \\ x[5,7] = abc \\ x[0,4] = \mathrm{Rec}^{-1}(g(abc)y)[8,12] \end{cases}$$

imply :

$$\begin{aligned} z_{13} + \mu &= a + (z_{12} + \lambda + \lambda')(z_{11} + 256\mu) + z_{10} + 256\lambda + 4352\lambda' + 2^{13} \\ z_{14} + 61681\lambda &= b + (z_{13} + \mu)(z_{12} + \lambda + \lambda') + (z_{11} + 256\mu) + 2^{14} \\ z_{15} + 61681\mu &= c + (z_{14} + 61681\lambda)(z_{13} + \mu) + (z_{12} + \lambda + \lambda') + 2^{15} \end{aligned}$$

where $z = \phi_R(abc)$. If we define :

$$\begin{aligned} a' &= a + z_{12} z_{11} + z_{10} + 2^{13} - z_{13} \\ b' &= b + z_{13} z_{12} + z_{11} + 2^{14} - z_{14} \\ c' &= c + z_{14} z_{13} + z_{12} + 2^{15} - z_{15} \end{aligned}$$

we have :

$$\begin{pmatrix} z_{11} + 256\mu + 256 & z_{11} + 256\mu + 4352 & a' - (1 - 256 z_{12})\mu \\ z_{13} + \mu - 61681 & z_{13} + \mu & b' + (256 + z_{12})\mu \\ 61681\,(z_{13} + \mu) + 1 & 1 & c' - (61681 - z_{14})\mu \end{pmatrix} \begin{pmatrix} \lambda \\ \lambda' \\ 1 \end{pmatrix} = 0$$

This is a linear system of unknown λ and λ'. If this system has an equation, which determinant has to be 0.

3.4 Discussion

This condition may be sufficient in most of the cases. The determinant should be a degree 3 polynomial. However, the coefficient of μ^3 is the determinant of the following matrix :

$$\begin{pmatrix} 256 & 256 & (1 - 256 z_{12}) \\ 1 & 1 & -(256 + z_{12}) \\ 61681 & 0 & (61681 - z_{14}) \end{pmatrix}$$

which is 0 since the first line is 256 time the second.

The coefficient of μ^2 is 0 with probability $1/p$, this is rare. In this case, we have one solution if the equation has a degree one, and zero or p solutions in the other cases.

μ has to satisfy a degree 2 equation. If the discriminant is different from 0, it has a square root with probability 50%. So, we have two different μ or no solution with probability 50%, and a single solution with probability $1/p$.

For each μ, we are likely to have a uniq solution (λ, λ'). However, it is possible to have 0 or p solutions, but it is rare. So, for each solution (λ, λ', μ), we can compute y in the system (3), then x. Finally, we have zero or two solutions x in almost all cases.

3.5 Reduction of the Function *FFT*

To sum up, we have a function f_R such that for a given abc :

$$f_R(abc) = \{D(x)[8,10]; x[5,7] = abc \wedge C(x)[11,15] = R\}$$

$f_R(abc)$ is a list of 0 or 2 $D(x)[8,10]$ for each x such that $x[5,7] = abc$ and $C(x)[11,15] = R$. The average of number of x is 1, so f_R is almost a function.

The function f_R is a kind of reduction of FFT since a collision for f_R gives a collision for FFT. We can use the birthday paradox with f_R to get some collision. The expected complexity is $O(2^{24})$.

We can invert FFT with f_R to. If we are looking for x such that $D(x)[8,15] = z$, we can compute $R = \text{Rec}^{-1}(z)[11,15]$ and look for abc such that $f_R(abc) = z[0,2]$. The complexity is 2^{48}. Then, we get the x required.

4 Finding Collisions with the Birthday Paradox

If we suppose that f_R is like a real random function, the probability that a set $\{f_R(x_i)\}$ for k different x_i have k elements is next to :

$$e^{-\frac{k^2}{2n}}$$

where n is the cardinality of the image of f_R, when k is next to $\sqrt{n}$. So, with $n = 2^{48}$ and $k = 3.2^{24}$, the probability is 1%.

Two collisions have been found in 24 hours by a SUN4 workstation with $k = 3.2^{24}$ different x. With the choice :

$$R = 5726\ 17fc\ b115\ c5c0\ a631$$

We got :

$$\begin{aligned} &FFT(17b3\ 2755\ 4e52\ b915\ 2218\ 1948\ 00a8\ 0002) = \\ &FFT(9c70\ 504e\ 834c\ b15c\ f404\ 94e2\ 02a7\ 0002) = \\ &\qquad 0851\ 393d\ 37c9\ 66e3\ d809\ d806\ 5e8c\ 05b8 \end{aligned}$$

and :

$$\begin{aligned} &FFT(8ccc\ 23a4\ 086d\ fbb9\ 85f4\ 70b2\ 029e\ 0002) = \\ &FFT(9d53\ 45ae\ 3286\ ada7\ 8c77\ 9877\ 02b4\ 0002) = \\ &\qquad 10e5\ 49f5\ 9df0\ d91b\ 0450\ afcc\ fba4\ 2063 \end{aligned}$$

Conclusion

The main weakness of FFT-Hash-II are described in section 2. First, the beginning of the computation depends on too few information of the input : $B(x)[0,7]$ is a function of $x[5,7]$. Second, the output allows to compute too much information of the computations in FFT : $D(x)[8,15]$ allows to compute $C(x)[11,15]$. The connection between $B(x)$ and $C(x)$ is linear, this makes our attack possible.

To get rid of the first weakness, we might mix H_0 and x in $A(x)$ before applying Rec. Similarly, the result of $< H_0, x >$ should be the set of $D(x)_{2i+1}$ instead of the right side.

Acknowledgment

I am happy to thank Jean-Marc Couveignes, Antoine Joux, Adi Shamir and Jacques Stern from the *Groupe de Recherche en Complexité et Cryptographie* for any advices. I owe a lot of time to Jacques Beigbeder, Ronan Keryell and all the *Service des Prestations Informatiques* for hardware and software advices. Finally, I should thank *France Telecom* to have given to me a phone number which hid so many collisions.

References

1. C. P. Schnorr. FFT-Hashing : An Efficient Cryptographic Hash Function. Presented at the rump session ot the CRYPTO'91 Conference (unpublished)
2. C. P. Schnorr. FFT-Hash II, Efficient Cryptographic Hashing. Presented at the EUROCRYPT'92 Conference (unpublished)
3. T. Baritaud, H. Gilbert, M. Girault. FFT Hashing is not Collision-free. Presented at the EUROCRYPT'92 Conference (unpublished)

This article was processed using the LaTeX macro package with LLNCS style