

# Cryptanalysis of a Provably Secure Cryptographic Hash Function

Jean-Sébastien Coron<sup>1</sup> and Antoine Joux<sup>2</sup>

<sup>1</sup> Gemplus Card International  
34 rue Guynemer, 92447 Issy-les-Moulineaux, France  
`jean-sebastien.coron@gemplus.com`

<sup>2</sup> DCSSI Crypto Lab  
51, Bd de Latour-Maubourg, 75700 Paris, France  
`antoine.joux@m4x.org`

**Abstract.** We present a cryptanalysis of a provably secure cryptographic hash function proposed by Augot, Finiasz and Sendrier in [1]. Our attack is a variant of Wagner’s generalized birthday attack. It is significantly faster than the attack considered in [1], and it is practical for two of the three proposed parameters.

## 1 Introduction

We describe a cryptanalysis of a provably secure cryptographic hash function proposed by Augot, Finiasz and Sendrier in [1]. The hash function is based on xoring the columns of a random binary matrix  $H$ , and is defined as follows:

**Initialization:** Let  $s = \omega \cdot a$  be the length of the input message, split into  $\omega$  blocks of  $a$  bits. Let  $r$  be the output size in bits. Let  $u = 2^a$ . Generate a random matrix  $H$  of  $r$  lines and  $n$  columns where  $n = \omega \cdot u$ . The matrix  $H$  is split into  $\omega$  sub-matrix  $H_i$  of size  $r \times u$ .

**Input:** a message  $m$  of  $s$  bits.

1. Split the  $s$  input bits in  $\omega$  parts  $s_1, \dots, s_\omega$  of  $a$  bits.
2. Convert each  $s_i$  into an integer between 1 and  $u = 2^a$ .
3. Choose the corresponding column in each sub-matrix  $H_i$ .
4. Xor the  $w$  chosen columns to obtain a  $r$ -bit string  $h$ .
5. Output the  $r$ -bit string  $h$ .

It is shown in [1] that the security of the hash function is reduced to the average case hardness of two NP-complete problems, namely the Regular Syndrome Decoding problem and the 2-Regular Null Syndrome Decoding problem.

The authors of [1] also describe an attack, called Information Set Decoding, and propose three set of parameters in order to make this attack unpractical.

The first set of parameters takes  $r = 160$ ,  $\omega = 64$ ,  $u = 256$ ,  $n = 2^{14}$  and has a conjectured security level of  $2^{62.3}$ . The second set of parameters takes  $r = 224$ ,  $\omega = 96$ ,  $u = 256$ ,  $n = 3 \cdot 2^{13}$  with a security level  $2^{82.3}$  and the third set of parameters takes  $r = 288$ ,  $\omega = 128$ ,  $u = 64$  and  $n = 2^{13}$ .

However, we describe in this paper a much faster attack, which is practical for the two first set of parameters.

## 2 Our Attack

### 2.1 Wagner's generalized birthday attack

Our attack is based on Wagner's generalized birthday attack [2], which is the following. Let  $L_1, \dots, L_4$  be four lists of  $n$ -bit random integers. The task is to find  $x_i \in L_i$  such that  $x_1 \oplus x_2 \oplus x_3 \oplus x_4 = 0$ . A solution exists with good probability if each list contains at least  $2^{n/4}$  integer. The obvious approach consists in generating all possible values of  $x_1 \oplus x_2$  and  $x_3 \oplus x_4$  and then look for a collision; this requires  $\mathcal{O}(2^{n/2})$  time.

Wagner's generalized birthday attack solves this problem in time  $\mathcal{O}(2^{n/3})$  for lists of size at least  $2^{n/3}$ . First one generates a list of roughly  $2^{n/3}$  values  $y = x_1 \oplus x_2$  such that the  $n/3$  low-order bits of  $y$  are zero. This can be done in time  $\mathcal{O}(2^{n/3})$ . The same is done for values  $z = x_3 \oplus x_4$ . One obtains two lists of roughly  $2^{n/3}$  integers with the  $n/3$  low-order bits set to zero. Then one looks for a collision between the two lists, and a solution is found in time  $\mathcal{O}(2^{n/3})$ .

This technique can be generalized to find a zero sum between  $2^a$  lists, and requires  $\mathcal{O}(2^a \cdot 2^{n/(a+1)})$  time with lists of size  $\mathcal{O}(2^{n/(a+1)})$ .

### 2.2 Our attack

Our attack against the previous hash function is then as follows. Our goal is to produce a collision, that is to produce two messages  $m \neq m'$  such that  $H(m) = H(m')$ . Therefore, for each of the  $\omega$  matrices  $H_i$  of  $u$  columns, we must select two columns, so that the xor of the  $2\omega$  columns gives 0.

For each sub-matrix  $H_i$ , we can generate a list  $L_i$  of roughly  $u^2/2$  values  $x_i$  which are the xor of 2 columns of  $H_i$ . Then we apply Wagner's algorithm to find a generalized birthday attack among the  $\omega$  lists:

$$x_1 \oplus x_2 \oplus \dots \oplus x_\omega = 0$$

More precisely, let  $\ell$  such that  $2^\ell = u^2/2$ . There are  $2^{2\ell}$  elements  $x_1 \oplus x_2$ , where  $x_1 \in L_1$  and  $x_2 \in L_2$ , among which  $2^\ell$  are such that the rightmost  $\ell$  bits are 0. This gives a list  $L'_1$ , which can be generated in time  $\mathcal{O}(2^\ell)$ . We can do the same with the lists  $(L_3, L_4)$  and obtain  $L'_2$ .

Then, by the birthday paradox, we can find an element in  $L'_1 \oplus L'_2$  with the  $3\ell$  rightmost bits equal to zero, in time  $\mathcal{O}(2^\ell)$ . Therefore, if  $\omega = 4$  and the hash size is  $r = 3\ell$ , we can find a collision in time  $\mathcal{O}(2^\ell)$ . We can generalize this to higher values of  $\omega$  by building the corresponding tree and we obtain that we can find a collision in time  $\mathcal{O}(\omega \cdot 2^\ell)$  if:

$$r \leq (\log_2(\omega) + 1) \cdot \ell$$

where  $\ell = 2\log_2(u) - 1$ .

Unfortunately, this is not enough for breaking the hash function for the recommended parameters, so we can generalize this by first taking all the  $2^{2\ell}$  elements

$x_1 \oplus x_2$ , and working with a tree with the same depth minus one. It is easy to see that one can find a collision in time  $\mathcal{O}(\omega \cdot 2^{2\ell})$  if :

$$r \leq 2(\log_2 \omega) \cdot \ell$$

This breaks the first instance with  $r = 160, \omega = 64, u = 256$  and  $\ell = 15$ , in time  $2^{36}$  (instead of  $2^{62}$  for the attack considered in the paper).

For the second instance ( $r = 224, \omega = 96, u = 256, \ell = 15$ ), we can first group the lists  $L_i$  by three, which gives 32 lists of  $2^{45}$  elements, from which we take only  $2^{38}$ . If  $\omega = 6$ , we can zero  $2 \cdot 38 = 76$  bits, if  $\omega = 12$ , we can zero  $3 \cdot 38 = 114$  bits, and with  $\omega = 96$ , we can zero  $6 \cdot 38 = 228$  bits, which breaks the hash function in time  $32 \cdot 2^{38} = 2^{43}$  (instead of  $2^{82}$  operations for the attack considered in the paper).

For the third instance ( $r = 288, w = 128, u = 64, \ell = 11$ ), we can group the lists  $L_i$  by six, and take  $2^{58}$  elements instead of  $2^{66}$ . With  $\omega = 12$ , we can zero  $2 \cdot 58 = 116$  bits, and with  $\omega = 96 < 128$ , we can zero  $5 \cdot 58 = 290$  bits, which breaks the hash function in time  $16 \cdot 2^{58} = 2^{62}$  (but this is probably not optimal).

### 3 Conclusion

We have described a cryptanalysis of a provably secure cryptographic hash function proposed by Augot, Finiasz and Sendrier in [1]. Our attack is a variant of Wagner's generalized birthday attack, and it is significantly faster than the attack considered in [1]. We have shown that it is practical for two of the three proposed parameters.

### References

1. D. Augot, M. Finiasz and N. Sendrier, *A Fast Provably Secure Cryptographic Hash Function*, Cryptology ePrint Archive, Report 2003/230. Available at <http://www.iacr.org/eprint>.
2. D. Wagner, *A Generalized Birthday Problem*, Proceedings of Crypto '02, LNCS vol. 2442, Springer-Verlage, 2002.