

# Re: CRC question

---

*Source:* <http://www.derkeiler.com/Newsgroups/sci.crypt/2006-08/msg01466.html>

---

- *From:* Francois GRIEU <fgrieu@xxxxxxxxxxxx>
  - *Date:* Thu, 24 Aug 2006 15:55:13 +0200
- 

In article <1156274464.334380.62710@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>, "hypermodest" <hypermodest@xxxxxxxx> wrote:

I have a source code of very special hash-function and I seek ways to crack it somehow. One part of this function is CRC-function

In the real world, in front of some function that you suspect is a CRC, but of unknown polynomial, and/or where you do not know the details such as modulus polynomial, initial and final message padding, order in which input bits form the divided polynomial, and/or the order in which outputs bit form the rest of polynomial division (or, in your case, fail to figure this out from code inspection), first thing is to check that a basic property of all CRC derivatives is verified:  
for any three messages A B C of same length  
 $H(A \text{ XOR } B \text{ XOR } C) = H(A) \text{ XOR } H(B) \text{ XOR } H(C)$

By this property alone,  $H(X)$  for any  $X$  of  $n$  bits can be trivially computed from the  $H(M_j)$  of only  $n+1$  appropriate fixed messages, such as the  $n+1$  distinct  $M_j$  of  $n$  bits with at most 1 bit set.

This technique works for a class of functions much wider than the textbook CRC, including all CRC-lookalikes for fixed length messages, without bothering for annoying details. I suspect this is enough for your needs.

if you figure out the order of bits in the messages and remainder, finding the reducing polynomial is relatively easy: independently of initial and final message padding value (but dependent on the number  $m$  of bits in the final padding, which is the order of the reducing polynomial in the textbook CRC, and in practice some small multiple of 8), the XOR of two distinct messages of same length,

Re: CRC question

left-shifted by  $m$  bits, then XORed with the CRC of each two messages, must be divisible by the reducing polynomial.

Thus, gather a few examples, find a polynomial GCD, and you get the reducing polynomial (or disprove your hypothesis on  $m$  or the order of bits).

François Grieru