Change List of JH

January 15, 2009

Hongjun Wu

Institute for Infocomm Research, Singapore
wuhongjun@gmail.com

1 Fixing a Bug in the Code

Paul Crowley has independently implemented the JH algorithms. He noticed that there is a bug in the "Update" function (this function is used in "jh_ref.h", "jh_opt32.h" and "jh_opt64.h"). The bug is that a partial block is always copied from the first message block into the buffer, so when the message length is larger than 512 bits, the partial block is copied wrongly.

In the original code, the partial block is copied as follows:

```
/*storing the partial block into buffer*/
if ( (databitlen & 0x1ff) > 0) {
       for (i = 0; i < 64; i++) state->buffer[i] = 0;
       if ((databitlen & 7) == 0) memcpy(state->buffer,
                     data, (databitlen & 0x1ff) >> 3);
       else memcpy(state->buffer,
                     data, ((databitlen & 0x1ff) >> 3)+1);
}
```

In the above code, the second parameter of memcpy is wrong. The second parameter of memcpy should be changed from "data" to "((databitlen $\gg$ 9) $\ll$ 6)" :

```
/*storing the partial block into buffer*/
if ( (databitlen & 0x1ff) > 0) {
for (i = 0; i < 64; i++) state->buffer[i] = 0;
      if ((databitlen & 7) == 0) memcpy(state->buffer,
          data+((databitlen >> 9) << 6),
          (databitlen & 0x1ff) >> 3);
      else memcpy(state->buffer,
          ((databitlen >> 9) << 6),
          ((databitlen & 0x1ff) >> 3)+1);
}
```

The above change fixes the bug in the code. The bug does not affect the test vectors of messages with length less than 512, and it does not affect the test vectors of message with length being multiple of 512 bits. The bug affects the test vectors of messages with length larger than 512 but not the multiple of 512.

2 Updating the document

The following changes are made to the document of JH. A number of typos are fixed.

1. In the 17th line in Section 1; (page 1)
   "involves around 600 active Sboxes" is changed to "involves more than 600 active Sboxes". Reason: Most of the differentials involve much more than 600 active Sboxes.

2. In the 4th line in Section 2; (page 1)
   "The input bits are divided into $\alpha \times 2^d$ elements" is changed to "The input bits are divided into $\prod_{i=0}^{d-1} \alpha_i$ elements"
   Reason: the latter gives a more general EDP since the number of elements may not be multiple of 2^d.

3. In the 4th line in Section 4.3.2; (page 6)
   " $b_{i+2^{N-1}} = a_{2i+1}$ " is changed to " $b_{i+2^{d-1}} = a_{2i+1}$ "
   Reason: N is a typo here. We can know that it is a typo by referring to Fig. 2.

4. In the 4th line in Section 4.3.3; (page 6)
   " for $i = 2^{d-1}$ to $2^d - 1$ " is changed to " for $i = 2^{d-2}$ to $2^{d-1} - 1$ "
   Reason: the value of $2i$ exceeds 2^d, so it is incorrect. We can get the correct value either from Fig. 3, or from the reference code.

5. In the 5th line in Section 4.3.3; (page 6)
   " for $i = 2^{d-1}$ to $2^d - 1$ " is changed to " for $i = 2^{d-2}$ to $2^{d-1} - 1$ "
   Reason: the same as above.

6. In the 13-th line in Section 4.4 (page 7)
   " if $C_r^{(d),r} = 0$, then $v_i = S_0(a_i)$ " is changed to " if $C_r^{(d),i} = 0$, then $v_i = S_0(a_i)$ "
   Reason: As we stated in Sec. 4.1, each constant bit is used to select Sboxes. Thus it is a typo here. $C_r^{(d),i}$ is the ith bit in the r-th round constant.

7. In the 2nd line in Section 5. (page 10)
   " 2^{d+1}-bit $H^{(i)}$ " is changed to " 2^{d+2}-bit $H^{(i)}$ "
   Reason: It is typo since it is already stated that $H^{(i-1)}$ is 2^{d+2} bits.

8. In the 3rd line in Section 6.1 (page 11)
   " $896-1-(\ell \bmod 512)$ zero bits" is changed to " $384-1+(-\ell \bmod 512)$ zero bits"
   Reason: The above two values are equal except when the message length is the multiple of 512. $384 - 1 + (-\ell \bmod 512)$ was suggested by Paul Crowley so that the padding in the document matches the padding in the code when the message length is a multiple of 512 bits. When the message length is multiple of 512 bits, only one block is to be padded, as given in the reference code. However, two blocks are padded to the message when the message length is the multiple of 512 if " $896 - 1 - (\ell \bmod 512)$ is used.

9. In the 2nd line in Section 6.5.1 (page 12)
   " $H^{(N),800} \| H^{(N),769} \| \cdots \| H^{(N),1023}$" is changed to
   " $H^{(N),800} \| H^{(N),801} \| \cdots \| H^{(N),1023}$"
   Reason: It is a typo. 800 is followed by 801.

10. In the 8th line in Section 9 (page 19)
    " $2^{512-\bar{l}}$ " is changed to " $2^{512-\log_2 \bar{l}}$ "
    Reason: It is a typo.

11. In the 3rd line in Section 10.1 (page 23)
    "512-bit memory for storing the memory" is changed to " 512-bit memory for storing the message block "
    Reason: typo

12. In the 2nd line in Section 11.2 (page 24)
    " a 2^d-dimensional array " is changed to "a d-dimensional array ".
    Reason: typo. d is used to indicate the dimension of the array in the document.

Acknowledgement. I would like to thank Paul Crowley for independently implementing JH, detecting the bug in my code and suggesting the formula for computing the number of zero bits being padded to the message.