Some Observations on HC-128

Subhamoy Maitra[1], Goutam Paul[2], Shashwat Raizada[1]

[1] Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, India.
subho@isical.ac.in, shashwat.raizada@gmail.com
[2] Department of Computer Science and Engineering,
Jadavpur University, Kolkata 700 032, India.
goutam_paul@cse.jdvu.ac.in

Abstract. In this paper, we use linear approximations of the addition modulo 2^n of three n-bit integers to identify linear approximations of g_1, g_2, the feedback functions of HC-128. This, in turn, shows that the process of keystream output generation of HC-128 can be well approximated by linear functions. In this direction, we show that the "least significant bit" based distinguisher (presented by the designer himself) of HC-128 works for the complete 32-bit word. In a different note, in the line of Dunkelman's observation, we also study how HC-128 keystream words leak secret state information of the cipher due to the properties of the functions h_1, h_2 and present improved results.

Keywords: Bias, Cryptography, Distinguishing Attack, eStream, Keystream, Linear Approximation, Stream Cipher.

1 Introduction

The eSTREAM [1] Portfolio (revision 1 in September 2008) contains the stream cipher HC-128 [6] in Profile 1 (SW). Apart from the analysis by the author (Wu) himself to conjecture the security of this cipher, the only other observation is by Dunkelman [2] in the eSTREAM discussion forum to show that the keystream words of HC-128 leak information regarding secret states. There is actually no other published result that shows any weakness of the cipher. In this paper, we identify a few other weaknesses of HC-128. Though our results do not constitute an attack on HC-128, we believe these will aid further exposure towards analysis of the cipher.

Each keystream word of HC-128 is 32 bit long (the 0th bit is the least significant bit and the 31st bit is the most significant bit). In [6], bitwise XOR of least significant bits of 10 (possibly) different keystream words (rotated by certain amounts) are considered to propose a distinguisher and it has been commented: "But due to the effect of the two '+' operations in the feedback function, the attack exploiting those 31 bits is not as effective as that exploiting the least significant bit". In Section 3, we discuss the linear approximation of the feedback functions g_1, g_2. These results are used in Section 4 to characterize the distinguisher for all other bits. In Section 4.2, we show that for each of the bits 2 to 31, one can have distinguishers of almost the same strength as the distinguisher proposed for the least significant bit in [6]. Thus it is shown that

- there are 30 many slightly weaker distinguishers other than the one described in [6] at bit level; these are based on biases of the order of 2^{-81};

– all these distinguishers can be taken together to mount a word level distinguisher for HC-128.

In Section 5, we study how the keystream output words leak secret state information in HC-128. In [2], it has been observed that "XOR of two consecutive keystream words of 32-bit each" is equal to the "XOR of two consecutive words of the secret array" with probability $\approx 2^{-16}$. We study this analysis in more detail and in the process we find a sharper association which gives twice the above probability.

We start with the description of HC-128 in the following section.

2 Description of HC-128

This is adapted from [6, Section 2].

2.1 Notations and Data Structures

The following operations are used in HC-128:

$+$: $x + y$ means $x + y \bmod 2^{32}$, where $0 \le x < 2^{32}$ and $0 \le y < 2^{32}$.
$\boxminus$: $x \boxminus y$ means $x - y \bmod 512$.
$\oplus$: bit-wise exclusive OR.
$\|$: concatenation.
$\gg$: right shift operator. $x \gg n$ means x being right shifted n bits.
$\ll$: left shift operator. $x \ll n$ means x being left shifted n bits.
$\ggg$: right rotation operator. $x \ggg n$ means $((x \gg n) \oplus (x \ll (32-n))$, where $0 \le n < 32$, $0 \le x < 2^{32}$.
$\lll$: left rotation operator. $x \lll n$ means $((x \ll n) \oplus (x \gg (32-n))$, where $0 \le n < 32$, $0 \le x < 2^{32}$.

Two tables P and Q, each with 512 many 32-bit elements are used as internal states of HC-128. A 128-bit key array $K[0, \ldots, 3]$ and a 128-bit initialization vector $IV[0, \ldots, 3]$ are used, where each entry of the array is a 32-bit element. Let s_t denote the keystream word generated at the t-th step, $t = 0, 1, 2, \ldots$.

The following six functions are used in HC-128:

$f_1(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3)$,
$f_2(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10)$,
$g_1(x, y, z) = ((x \ggg 10) \oplus (z \ggg 23)) + (y \ggg 8)$,
$g_2(x, y, z) = ((x \lll 10) \oplus (z \lll 23)) + (y \lll 8)$,
$h_1(x) = Q[x^{(0)}] + Q[256 + x^{(2)}]$,
$h_2(x) = P[x^{(0)}] + P[256 + x^{(2)}]$,

where $x = x^{(3)} \| x^{(2)} \| x^{(1)} \| x^{(0)}$, x is a 32-bit word and $x^{(0)}$ (least significant byte) $, x^{(1)}, x^{(2)}$ and $x^{(3)}$ (most significant byte) are four bytes.

2.2 Key and IV Setup

1. Let $K[0,\ldots,3]$ be the secret key and $IV[0,\ldots,3]$ be the initialization vector. Let $K[i+4] = K[i]$ and $IV[i+4] = IV[i]$ for $0 \le i \le 3$.
2. The key and IV are expanded into an array $W[0,\ldots,1279]$ as follows.

$$W[i] = \begin{cases} K[i] & 0 \le i \le 7; \\ IV[i-8] & 8 \le i \le 15; \\ f_2(W[i-2]) + W[i-7] + f_1(W[i-15]) + W[i-16] + i & 16 \le i \le 1279. \end{cases}$$

3. Update the tables P and Q with the array W as follows.

$$P[i] = W[i+256], \text{ for } 0 \le i \le 511$$
$$Q[i] = W[i+768], \text{ for } 0 \le i \le 511$$

4. Run the cipher 1024 steps and use the outputs to replace the table elements as follows.
   for $i = 0$ to 511, do
   $$P[i] = (P[i] + g_1(P[i \boxminus 3], P[i \boxminus 10], P[i \boxminus 511])) \oplus h_1(P[i \boxminus 12]);$$
   for $i = 0$ to 511, do
   $$Q[i] = (Q[i] + g_2(Q[i \boxminus 3], Q[i \boxminus 10], Q[i \boxminus 511])) \oplus h_2(Q[i \boxminus 12]);$$

2.3 The Keystream Generation Algorithm

$i = 0$;
repeat until enough keystream bits are generated
{
  $j = i \bmod 512$;
  if $(i \bmod 1024) < 512$
  {
    $P[j] = P[j] + g_1(P[j \boxminus 3], P[j \boxminus 10], P[j \boxminus 511])$;
    $s_i = h_1(P[j \boxminus 12]) \oplus P[j]$;
  }
  else
  {
    $Q[j] = Q[j] + g_2(Q[j \boxminus 3], Q[j \boxminus 10], Q[j \boxminus 511])$;
    $s_i = h_2(Q[j \boxminus 12]) \oplus Q[j]$;
  }
  end-if
  $i = i + 1$;
}
end-repeat

3 Linear Approximation of Feedback Functions g_1, g_2

HC-128 uses two functions g_1, g_2 of similar kind. The two '+' operations in g_1 or g_2 are believed to be a source of high nonlinearity, but we find good linear approximation in this case by using the result of linear approximation of the addition of three integers.

Linear approximations of modulo-2^n addition of k many n-bit integers have been studied in [5]. For $k = 2$, the probability of the equality of XOR and modulo-2^n sum in the i-th least significant bit tends to $\frac{1}{2}$ as i increases. Below, we briefly discuss the case for $k = 3$, i.e., the XOR-approximation of modulo addition of three integers, that would be subsequently used in approximating g_1, g_2. We do not claim that the probability calculation in Theorem 1 below as our contribution, but we have presented an outline for better understanding.

Let X_1, X_2, X_3 be three n-bit integers; $S = (X_1 + X_2 + X_3) \bmod 2^n$, $T = X_1 \oplus X_2 \oplus X_3$, the bitwise XOR. For $i \geq 0$, let C_i denote the carry produced in the i-th step of the addition of X_1, X_2 and X_3. Since three bits are involved, C_i can take the values $0, 1$ and 2. For the LSB addition, we assume $C_{-1} = 0$. Denote $p_{i,v} = Prob(C_i = v)$, $i \geq -1$, $v \in \{0, 1, 2\}$. We know that $Prob(S_i = T_i) = Prob(C_{i-1} = 0 \text{ or } 2) = p_{i-1,0} + p_{i-1,2} = 1 - p_{i-1,1}$. The following recurrences are easy to show.

1. $p_{i+1,0} = \frac{1}{2}p_{i,0} + \frac{1}{8}p_{i,1}$.
2. $p_{i+1,1} = \frac{1}{2}p_{i,0} + \frac{3}{4}p_{i,1} + \frac{1}{2}p_{i,2}$.
3. $p_{i+1,2} = \frac{1}{8}p_{i,1} + \frac{1}{2}p_{i,2}$.

The solution gives $p_{i,1} = \frac{2}{3}(1 - \frac{1}{4^{i+1}})$ and so we have the following result.

Theorem 1. *For $i \geq 0$, $Prob(S_i = T_i) = \frac{1}{3}(1 + \frac{1}{2^{2i-1}})$.*

As we will be using the keystream word number as subscript, we will denote the b-th least significant bit of an n-bit word w by w^b, $0 \leq b \leq n-1$, i.e., $w = (w^{n-1}, w^{n-2}, \ldots, w^1, w^0)$. This notation is also extended to w^b, where $b > n-1$. In that case, w^b will mean $w^{b \bmod n}$.

Based on this notation and using approximation to Theorem 1, we write the following result.

Corollary 1. *Suppose X_1, X_2, X_3 are three n-bit integers with $\mathcal{S} = (X_1 + X_2 + X_3) \bmod 2^n$. Then, for $0 \leq b \leq n-1$,*

$$Prob(\mathcal{S}_i^b = X_1^b \oplus X_2^b \oplus X_3^b) = p_b,$$

where $p_b = \frac{1}{3}(1 + \frac{1}{2^{2b-1}})$, i.e.,

$$p_b = \begin{cases} 1 & \text{if } b = 0; \\ \frac{1}{2} & \text{if } b = 1; \\ \frac{1}{3} \text{ (approximately)} & \text{if } 2 \leq b \leq n-1. \end{cases}$$

During the keystream generation part of HC-128, the array P is updated as

$$P[i] = P[i] + g_1(P[i \boxminus 3], P[i \boxminus 10], P[i \boxminus 511]),$$

where

$$g_1(x, y, z) = ((x \ggg 10) \oplus (z \ggg 23)) + (y \ggg 8).$$

Thus, the update rule can be restated as

$$P_{updated}[i] = P[i] + \big((P[i \boxminus 3] \ggg 10) \oplus (P[i \boxminus 511] \ggg 23)\big) + (P[i \boxminus 10] \ggg 8).$$

Suppose $P'_{updated}$ is the updated value of $P[i]$, when we replace the two $+$'s by $\oplus$'s in the right hand side. Then for $0 \leq b \leq n-1$, the b-th bit of the updated value would be given by

$$(P'_{updated}[i])^b = (P[i])^b \oplus (P[i \boxminus 3])^{10+b} \oplus (P[i \boxminus 511])^{23+b} \oplus (P[i \boxminus 10])^{8+b}.$$

According to Corollary 1, for $0 \leq b \leq n-1$, we have

$$Prob\big((P'_{updated}[i])^b = (P_{updated}[i])^b\big) = p_b.$$

Following the same notation as in [6, Section 4], we may write the keystream generation step as

$$s_i = h_1(P_{updated}[i \boxminus 12]) \oplus P_{updated}[i],$$

for $0 \leq i \bmod 1024 < 512$. Consider

$$\psi_i^b = \begin{cases} (h_1(P_{updated}[i \boxminus 12]) \oplus P'_{updated}[i])^b & \text{if } b = 0, 1; \\ 1 \oplus (h_1(P_{updated}[i \boxminus 12]) \oplus P'_{updated}[i])^b & \text{if } 2 \leq b < 32. \end{cases}$$

Then we have the following result.

Theorem 2. *The expected number of bits where the two 32-bit integers s_i and ψ_i match is* 21.5.

Proof. Let $m_b = 1$, if $s_i^b = \psi_i^b$; otherwise, let $m_b = 0$, $0 \leq b \leq 31$. Hence, the total number of matches is given by $M = \sum_{b=0}^{31} m_b$. By linearity of expectation, $E(M) = \sum_{b=0}^{31} E(m_b) = \sum_{b=0}^{31} Prob(m_b = 1)$. The probabilities $Prob(m_b = 1)$ can be computed from Corollary 1. As $\psi_i^b = 1 \oplus (h_1(P[i \boxminus 12]) \oplus P'_{updated}[i])^b$ for $2 \leq b < 32$, in these cases $Prob(s_i^b = \psi_i^b) \approx \frac{2}{3}$. Further, $Prob(s_i^1 = \psi_i^1) = \frac{1}{2}$ and $Prob(s_i^0 = \psi_i^0) = 1$. This gives the value of $E(M)$ as $30 \cdot \frac{2}{3} + \frac{1}{2} + 1 = 21.5$. □

Theorem 2 shows the association of the HC-128 keystream words s_i with its linear approximation ψ_i.

4 A Class of Distinguishers for HC-128

In this section, we use the linear approximation of the feedback functions g_1, g_2 described in the previous section to construct 30 new bit-level distinguishers.

4.1 Brief Outline of the Distinguisher of [6]

Before presenting the ideas in this section, let us revisit the keystream word generation of HC-128. The keystream words are generated using both the arrays P, Q, each consisting of 512 many words. However, the updates of P and Q arrays are independent. For 512 many iterations, the array P is updated with the older values from P itself and for the next 512 many iterations the array Q is updated with the older values of Q as well and this continues alternatively. Below, while discussing the distinguisher, following the notations of [6], we consider the older P and the updated P at the same time as an array of 1024 many words. The idea is mentioned more clearly with the following tabular representation. This is how the keystream words s_i's are related to the array elements $P[i]$'s for explaining the distinguisher.

...

Old P array:	$P[t+0]$	$P[t+1]$	...	$P[t+i-512]$	...	$P[t+511]$
Keystream:	s_t	s_{t+1}	...	$s_{t+i-512}$	...	s_{t+511}
Intermediate Q array:	$Q[t+0]$	$Q[t+1]$	...	...	...	$Q[t+511]$
Keystream:	s_{t+512}	s_{t+513}	...	...	...	s_{t+1023}
New P array:	$P[t+512]$	$P[t+513]$	...	$P[t+i]$	...	$P[t+1023]$
Keystream:	s_{t+1024}	s_{t+1025}	...	$s_{t+i+512}$	...	s_{t+1535}

...

Table 1. Evolution of the Array P and Correspondence with the Keystream Words s_i's.

Thus, for $10 \leq i < 511$, $P[i]$ (before the update in step i) corresponds to s_{i-1024} and $P[i \boxminus 511]$ corresponds to s_{i-1023}. The keystream output word of HC-128 is generated as $s_i = h_1(P[i \boxminus 12]) \oplus P[i]$, following an update of $P[i]$ by adding to it $g_1(P[i \boxminus 3], P[i \boxminus 10], P[i \boxminus 511])$, $0 \leq i \bmod 1024 < 512$. In other words, we can ignore the update and write the keystream generation as follows:

$$s_i = h_1(P[i \boxminus 12]) \oplus \big(P[i] + g_1(P[i \boxminus 3], P[i \boxminus 10], P[i \boxminus 511])\big)$$

or

$$s_i \oplus h_1(P[i \boxminus 12]) = P[i] + g_1(P[i \boxminus 3], P[i \boxminus 10], P[i \boxminus 511]).$$

Denoting $P[i \boxminus 12]$ at the i-th step as z_i, and substituting $P[i] = s_i \oplus h_1(z_i)$ in the update rule for P, we get, for $10 \leq i \bmod 1024 < 511$,

$$s_i \oplus h_1(z_i) = \big(s_{i-1024} \oplus h_1'(z_{i-1024})\big) + g_1\big(s_{i-3} \oplus h_1(z_{i-3}), s_{i-10} \oplus h_1(z_{i-10}), s_{i-1023} \oplus h_1'(z_{i-1023})\big).$$

Here $h_1(.)$ and $h_1'(.)$ indicate two different functions since they are related to two P arrays at two different 1024 size blocks that act as two different S-boxes. Inside g_1, we have three rotations, one XOR and one addition and outside g_1 we have one more addition. Since the LSB of the XOR of two words equal the LSB of the sum of those two words, we can write the above equation as

$$s_i^0 \oplus s_{i-1024}^0 \oplus s_{i-3}^{10} \oplus s_{i-10}^8 \oplus s_{i-1023}^{23}$$

$$= h_1(z_i)^0 \oplus h_1'(z_{i-1024})^0 \oplus h_1(z_{i-3})^{10} \oplus h_1(z_{i-10})^8 \oplus h_1'(z_{i-1023})^{23}.$$

Thus, for $1024\tau + 10 \leq j < i < 1024\tau + 511$,

$$s_i^0 \oplus s_{i-1024}^0 \oplus s_{i-3}^{10} \oplus s_{i-10}^8 \oplus s_{i-1023}^{23}$$

$$= s_j^0 \oplus s_{j-1024}^0 \oplus s_{j-3}^{10} \oplus s_{j-10}^8 \oplus s_{j-1023}^{23}$$

if and only if $H(Z_i) = H(Z_j)$, where

$$H(Z_i) = h_1(z_i)^0 \oplus h_1'(z_{i-1024})^0 \oplus h_1(z_{i-3})^{10} \oplus h_1(z_{i-10})^8 \oplus h_1'(z_{i-1023})^{23}.$$

Here $Z_i = (z_i, z_{i-1024}, z_{i-3}, z_{i-10}, z_{i-1023})$ is an 80-bit input and $H(.)$ can be assumed as a random 80-bit-to-1-bit S-box.

The following result (with proof for better clarity) gives the collision probability for a general random m-bit-to-n-bit S-box.

Proposition 1. *[6, Theorem 1] Let H be an m-bit-to-n-bit S-box and all those n-bit elements are randomly generated, where $m \geq n$. Let x_1 and x_2 be two m-bit random inputs to H. Then $H(x_1) = H(x_2)$ with probability $2^{-m} + 2^{-n} - 2^{-m-n}$.*

Proof. If $x_1 = x_2$ (this happens with probability 2^{-m}), then $H(x_1) = H(x_2)$ happens with probability 1. If $x_1 = x_2$ (this happens with probability $1 - 2^{-m}$), then $H(x_1) = H(x_2)$ happens with probability 2^{-n}. Thus, $Prob\big(H(x_1) = H(x_2)\big) = 2^{-m} \cdot 1 + (1 - 2^{-m}) \cdot 2^{-n}$. □

Coming back to HC-128, $m = 80$ and $n = 1$ for the S-box whose outputs are $H(Z_i)$ and $H(Z_j)$, we have, according to Proposition 1, $Prob\big(H(Z_i) = H(Z_j)\big) = \frac{1}{2} + 2^{-81}$. Hence, for $1024\tau + 10 \leq j < i < 1024\tau + 511$,

$$Prob\Big(s_i^0 \oplus s_{i-1024}^0 \oplus s_{i-3}^{10} \oplus s_{i-10}^8 \oplus s_{i-1023}^{23} = s_j^0 \oplus s_{j-1024}^0 \oplus s_{j-3}^{10} \oplus s_{j-10}^8 \oplus s_{j-1023}^{23}\Big) = \frac{1}{2} + 2^{-81}.$$

Thus, a distinguisher can be mounted based on the equality of the least significant bits of the keystream word combinations $s_i \oplus s_{i-1024} \oplus (s_{i-3} \ggg 10) \oplus (s_{i-10} \ggg 8) \oplus (s_{i-1023} \ggg 23)$ and $s_j \oplus s_{j-1024} \oplus (s_{j-3} \ggg 10) \oplus (s_{j-10} \ggg 8) \oplus (s_{j-1023} \ggg 23)$. According to [6, Section 4], this distinguisher requires 2^{164} pairs of above keystream word combinations for a success probability 0.9772. It has been commented in [6] that the distinguisher will not be effective due to the use of modulo addition. In contrary to the belief of the designer of HC-128, we show in the next section that the distinguisher works for all the bits (except one) in the keystream words.

4.2 Extending the Distinguisher of [6] to Other Bits

Our analysis shows that there exist biases in the equality of 31 out of the 32 bits (except the second least significant bit) of the word combinations $s_i \oplus s_{i-1024} \oplus (s_{i-3} \ggg 10) \oplus (s_{i-10} \ggg 8) \oplus (s_{i-1023} \ggg 23)$ and $s_j \oplus s_{j-1024} \oplus (s_{j-3} \ggg 10) \oplus (s_{j-10} \ggg 8) \oplus (s_{j-1023} \ggg 23)$, which leads to a distinguisher for each of those 31 bits separately.

Our analysis generalizes the idea of [6, Section 4] by applying Corollary 1. We refer to the visualization of the array P as explained in Table 1. The keystream output word of HC-128 is generated as $s_i = h_1(P[i \boxminus 12]) \oplus P[i]$, $0 \le i \bmod 1024 < 512$. Denoting $P[i \boxminus 12]$ at the i-th step as z_i, and substituting $P[i] = s_i \oplus h_1(z_i)$ in the update rule for P, we get, for $10 \le i \bmod 1024 < 511$,

$$s_i \oplus h_1(z_i) = \big(s_{i-1024} \oplus h'_1(z_{i-1024})\big) + g_1\big(s_{i-3} \oplus h_1(z_{i-3}), s_{i-10} \oplus h_1(z_{i-10}), s_{i-1023} \oplus h'_1(z_{i-1023})\big).$$

Here $h_1(.)$ and $h'_1(.)$ indicate two different functions since they are related to two P arrays at two different 1024 size blocks that act as two different S-boxes.

As per the discussion following Corollary 1, we can write, for $10 \le i \bmod 1024 < 511$,

$$\left.\begin{array}{c} s_i^b \oplus s_{i-1024}^b \oplus s_{i-3}^{10+b} \oplus s_{i-10}^{8+b} \oplus s_{i-1023}^{23+b} \\ = h_1(z_i)^b \oplus h'_1(z_{i-1024})^b \oplus h_1(z_{i-3})^{10+b} \oplus h_1(z_{i-10})^{8+b} \oplus h'_1(z_{i-1023})^{23+b} \end{array}\right\} \quad (1)$$

holds with probability $p_0 = 1$ for $b = 0$, with probability $p_1 = \frac{1}{2}$ for $b = 1$ and with probability $p_b = \frac{1}{3}$ for $2 \le b \le 31$. In short, we can write, for $0 \le b \le 31$,

$$Prob\big(\Psi_i^b = H_b(Z_i)\big) = p_b,$$

where

$$\Psi_i^b = s_i^b \oplus s_{i-1024}^b \oplus s_{i-3}^{10+b} \oplus s_{i-10}^{8+b} \oplus s_{i-1023}^{23+b}$$

and

$$H_b(Z_i) = h_1(z_i)^b \oplus h'_1(z_{i-1024})^b \oplus h_1(z_{i-3})^{10+b} \oplus h_1(z_{i-10})^{8+b} \oplus h'_1(z_{i-1023})^{23+b}.$$

Here $Z_i = (z_i, z_{i-1024}, z_{i-3}, z_{i-10}, z_{i-1023})$ is an 80-bit input and each $H_b(.)$, $0 \le b \le 31$, is a random 80-bit-to-1-bit S-box. Obviously, for $0 \le b \le 31$, $Prob\big(\Psi_i^b = H_b(Z_i) \oplus 1\big) = 1 - p_b$.

Thus, we can state the following technical result.

Lemma 1. *For* $1024\tau + 10 \le j < i < 1024\tau + 511$ *and* $0 \le b \le 31$,

$$Prob\big(\Psi_i^b \oplus \Psi_j^b = H_b(Z_i) \oplus H_b(Z_j)\big) = q_b$$

where

$$q_b = \begin{cases} 1 \text{ if } b = 0; \\ \frac{1}{2} \text{ if } b = 1; \\ \frac{5}{9} \text{ if } 2 \le b \le 31. \end{cases}$$

Proof. $Prob\big(\Psi_i^b \oplus \Psi_j^b = H_b(Z_i) \oplus H_b(Z_j)\big)$
$= Prob\big(\Psi_i^b = H_b(Z_i)\big) \cdot Prob\big(\Psi_j^b = H_b(Z_j)\big) + Prob\big(\Psi_i^b = H_b(Z_i) \oplus 1\big) \cdot Prob\big(\Psi_j^b = H_b(Z_j) \oplus 1\big)$
$= p_b \cdot p_b + (1 - p_b) \cdot (1 - p_b)$.
Substituting the values of p_b from Corollary 1, we get the result. □

Obviously, for $0 \le b \le 31$, $Prob\big(\Psi_i^b \oplus \Psi_j^b = H_b(Z_i) \oplus H_b(Z_j) \oplus 1\big) = 1 - q_b$.

For a given b, all the $H_b(Z_i)$'s are the outputs of the same random secret 80-bit-to-1-bit S-box $H_b(.)$. So setting $m = 80$ and $n = 1$ in Proposition 1, we get the following corollary.

Corollary 2. *For* $1024\tau + 10 \leq j < i < 1024\tau + 511$ *and* $0 \leq b \leq 31$,

$$Prob\big(H_b(Z_i) = H_b(Z_j)\big) = \frac{1}{2} + 2^{-81}.$$

Obviously, $Prob\big(H_b(Z_i) = H_b(Z_j) \oplus 1\big) = \frac{1}{2} - 2^{-81}$.

Combining the above results, we get the following theorem.

Theorem 3. *For* $1024\tau + 10 \leq j < i < 1024\tau + 511$, $Prob(\Psi_i^b = \Psi_j^b) = \rho_b$, *where*

$$\rho_b = \begin{cases} \frac{1}{2} + 2^{-81} & \text{if } b = 0; \\ \frac{1}{2} & \text{if } b = 1; \\ \frac{1}{2} + \frac{2^{-81}}{9} & \text{if } 2 \leq b \leq 31. \end{cases}$$

Proof. $Prob(\Psi_i^b = \Psi_j^b)$
$= Prob\big(\Psi_i^b \oplus \Psi_j^b = H_b(Z_i) \oplus H_b(Z_j)\big) \cdot Prob\big(H_b(Z_i) = H_b(Z_j)\big) + Prob\big(\Psi_i^b \oplus \Psi_j^b = H_b(Z_i) \oplus H_b(Z_j) \oplus 1\big) \cdot Prob\big(H_b(Z_i) = H_b(Z_j) \oplus 1\big)$.
Substituting values from Lemma 1 and Corollary 2, we get the result. □

Note that for the special case of $b = 0$, we have a distinguisher based on the bias $\frac{1}{2} + 2^{-81}$ in the equality of the LSB's of Ψ_i and Ψ_j. This is exactly the distinguisher described in [6, Section 4]. Our results show that we can also mount a distinguisher of around the same order for each of the 30 bits corresponding to $b = 2, 3, \ldots, 31$ based on the bias $\frac{1}{2} + \frac{2^{-81}}{9}$.

If one checks how many bit positions match between two random 32-bit integers, the expected value is 16. Below we show that if one performs a bitwise comparison of the 32-bit elements $\Psi_i = (\Psi_i^{31}, \Psi_i^{30}, \ldots, \Psi_i^0)$ and $\Psi_j = (\Psi_j^{31}, \Psi_j^{30}, \ldots, \Psi_j^0)$ in HC-128, where $1024\tau + 10 \leq j < i < 1024\tau + 511$, then the expected number of matches between the corresponding bits is more than 16, and to be precise, is $16 + \frac{13}{12} \cdot 2^{-79}$.

Theorem 4. *For* $1024\tau + 10 \leq j < i < 1024\tau + 511$, *the expected number of bits where the two 32-bit integers* Ψ_i *and* Ψ_j *match is* $16 + \frac{13}{12} \cdot 2^{-79}$.

Proof. Let $m_b = 1$, if $\Psi_i^b = \Psi_j^b$; otherwise, let $m_b = 0$, $0 \leq b \leq 31$. Hence, the total number of matches is given by $M = \sum_{b=0}^{31} m_b$. From Theorem 3, we have $Prob(m_b = 1) = \rho_b$. Hence, $E(m_b) = \rho_b$ and by linearity of expectation, $E(M) = \sum_{b=0}^{31} E(m_b) = \sum_{b=0}^{31} \rho_b$. Substituting the values of ρ_b's from Theorem 3, we get $E(M) = 16 + \frac{13}{3} \cdot 2^{-81}$. □

Thus our contributions in this section constitute of

- identifying 30 many slightly weaker distinguishers other than the one described in [6] at bit level (Theorem 3);
- further, all these distinguishers can be taken together to mount a word level distinguisher for HC-128 (Theorem 4).

These distinguishers have not been identified in [6].

5 Collisions in h_1, h_2 and State Leakage in Keystream

Whereas the previous sections concentrated on the functions g_1, g_2; here, in a different direction, we study the other two functions h_1, h_2. Without loss of generality, we focus on the keystream block corresponding to the P array, i.e., the block of 512 rounds where P is updated in each round and Q remains constant. As j runs from 0 to 511, we denote the corresponding output $h_1(P[j \boxminus 12]) \oplus P[j]$ by s_j. Here, $h_1(x) = Q[x^{(0)}] + Q[256 + x^{(2)}]$. The results we present in this section are in terms of the function h_1. The same analysis holds for the function h_2 in the other keystream block.

In [2], it has been observed that $Prob(s_j \oplus s_{j+1} = P[j] \oplus P[j+1]) \approx 2^{-16}$, where s_j, s_{j+1} are two consecutive keystream output words. We study that in more detail in this section and in the process we find a sharper association in Theorem 6 which gives twice the above probability.

The following technical result establishes that XOR of two words of P is leaked in the keystream words if the corresponding values of $h_1(.)$ collide.

Lemma 2. *For $0 \leq u \neq v \leq 511$, $s_u \oplus s_v = P[u] \oplus P[v]$ if and only if $h_1(P[u \boxminus 12]) = h_1(P[v \boxminus 12])$.*

Proof. We have $s_u = h_1(P[u \boxminus 12]) \oplus P[u]$ and $s_v = h_1(P[v \boxminus 12]) \oplus P[v]$. Thus, $s_u \oplus s_v = (h_1(P[u \boxminus 12]) \oplus h_1(P[v \boxminus 12])) \oplus (P[u] \oplus P[v])$. The term $(h_1(P[u \boxminus 12]) \oplus h_1(P[v \boxminus 12]))$ vanishes if and only if $s_u \oplus s_v = P[u] \oplus P[v]$. □

Now we detail the result related to collision in h_1. Note that the array P from which the input to the function h_1 is selected and the array Q from which the output of h_1 is chosen can be assumed to contain uniformly distributed 32-bit elements. In Lemma 3, which is in a more general setting than just HC-128, we use notations h and U; these may be considered to model h_1 and Q respectively.

Lemma 3. *Let $h(x) = U[x^{(0)}] + U[x^{(2)} + 2^m]$ be an n-bit to n-bit mapping, where each entry of the array U is an n-bit number, U contains 2^{m+1} many elements and $x^{(0)}$ and $x^{(2)}$ are two disjoint m-bit segments from the n-bit input x. Suppose x and x' are two n-bit random inputs to h. Assuming that the entries of U are distributed uniformly at random, we have $Prob\big(h(x) = h(x')\big) = \alpha_{m,n}$, where*

$$\alpha_{m,n} = 2^{-2m} + 2^{1-m-n}(1 - 2^{-m}) + 2^{-2n}(1 - 2^{-m})^2 + 2^{-n}(1 - 2^{-m})^2(1 - 2^{-n})^2.$$

Proof. The value of $h(x)$ equals the value $h(x')$ in the following five ways.

1. $x^{(0)} = x'^{(0)}$ and $x^{(2)} = x'^{(2)}$. This happens with probability $2^{-m} \cdot 2^{-m}$.
2. $x^{(0)} = x'^{(0)}$ and $x^{(2)} \neq x'^{(2)}$ and $U[x^{(2)}] = U[x^{(2)}]$. This happens with probability $2^{-m} \cdot (1 - 2^{-m}) \cdot 2^{-n}$.
3. $x^{(0)} \neq x'^{(0)}$ and $x^{(2)} = x'^{(2)}$ and $U[x^{(0)}] = U[x'^{(0)}]$. This happens with probability $2^{-m} \cdot (1 - 2^{-m}) \cdot 2^{-n}$.
4. $x^{(0)} \neq x'^{(0)}$ and $x^{(2)} \neq x'^{(2)}$ and $U[x^{(0)}] = U[x'^{(0)}]$ and $U[x^{(2)}] = U[x'^{(2)}]$. This happens with probability $(1 - 2^{-m}) \cdot (1 - 2^{-m}) \cdot 2^{-n} \cdot 2^{-n}$.

5. $x^{(0)} \neq x'^{(0)}$ and $x^{(2)} \neq x'^{(2)}$ and $U[x^{(0)}] \neq U[x'^{(0)}]$ and $U[x^{(2)}] \neq U[x'^{(2)}]$, but still $h(x) = h(x')$ due to random association. This happens with probability $(1-2^{-m}) \cdot (1-2^{-m}) \cdot (1-2^{-n}) \cdot (1-2^{-n}) \cdot 2^{-n}$.

Adding the above five components, we get the result. □

The following corollary comes from Lemma 3 when we consider any t out of n bits. The notation $x =_t y$ means x and y match in any predefined set of t bits, $0 \leq t \leq n$.

Corollary 3. *For $0 \leq t \leq n$, we have $Prob\big(h(x) =_t h(x')\big) = p_t$, where*

$$p_{m,n,t} = \alpha_{m,n} + (1-\alpha_{m,n})2^{-t}.$$

Proof. The event $\big(h(x) =_t h(x')\big)$ can occur in the following two ways.

1. When $h(x) = h(x')$ and thus any t-bit portions are also equal. According to Lemma 3, this happens with probability $\alpha_{m,n}$.
2. When $h(x) \neq h(x')$, the two fixed t-bit segments may equal due to random association. This happens with probability $(1-\alpha_{m,n})2^{-t}$.

Adding the two components, we get the result. □

Note that $\alpha_{m,n} > 2^{-2m}$ and the main contributing part to $\alpha_{m,n}$ is 2^{-2m} (see item 1 in the proof of Lemma 3) when $m < \frac{n}{2}$. For HC-128, $m = \frac{n}{4}$ and that creates a bias in the equality of $h_1(.)$ for two different inputs. With $m = 8$ and $n = 32$, the above probability turns out to be $\alpha_{8,32} = 0.0000152590$ which is slightly greater than 2^{-16}. We like to point out that if one checks the equality of two n-bit random integers, then the probability of that event is 2^{-n} only, which is as low as 2^{-32}.

Next we formalize the result given in [2].

Theorem 5. *In HC-128, consider a block of 512 many keystream words corresponding to array P. For $0 \leq u \neq v \leq 511$, $Prob\big((s_u \oplus s_v) = (P[u] \oplus P[v])\big) = \alpha_{8,32} > 2^{-16}$.*

Proof. The result follows from Lemma 2 and Lemma 3. □

Now, we present a sharper result which gives twice the probability of the observation in [2].

Theorem 6. *In HC-128, consider a block of 512 many keystream words corresponding to array P. For any u, v, $0 \leq u \neq v < 500$, if $\big((s_u^{(0)} = s_v^{(0)}) \;\&\; (s_u^{(2)} = s_v^{(2)})\big)$, then*

$$Prob\big((s_{u+12} \oplus s_{v+12}) = (P[u+12] \oplus P[v+12])\big) \approx \frac{1}{2^{15}}.$$

Proof. From Lemma 2, $s_u^{(b)} \oplus s_v^{(b)} = P[u]^{(b)} \oplus P[v]^{(b)}$ if and only if $h_1(P[u \boxminus 12])^{(b)} = h_1(P[v \boxminus 12])^{(b)}$, for $b = 0, 1, 2, 3$. Given that $s_u^{(0)} = s_v^{(0)}$ and $s_u^{(2)} = s_v^{(2)}$, we have
$P[u]^{(0)} = P[v]^{(0)}$ and $P[u]^{(2)} = P[v]^{(2)}$ if and only if
$h_1(P[u \boxminus 12])^{(0)} = h_1(P[v \boxminus 12])^{(0)}$ and $h_1(P[u \boxminus 12])^{(2)} = h_1(P[v \boxminus 12])^{(2)}$.

Thus,
$Prob\Big(P[u]^{(0)} = P[v]^{(0)} \ \&\ P[u]^{(2)} = P[v]^{(2)} \mid s_u^{(0)} = s_v^{(0)} \ \&\ s_u^{(2)} = s_v^{(2)}\Big)$
$= Prob\Big(h_1(P[u \boxminus 12])^{(0)} = h_1(P[v \boxminus 12])^{(0)} \ \&\ h_1(P[u \boxminus 12])^{(2)} = h_1(P[v \boxminus 12])^{(2)}\Big)$
$= p_{8,32,16} \approx \alpha_{8,32} + (1 - \alpha_{8,32})2^{-16}$ (from Corollary 3)
$\approx \frac{1}{2^{15}}$.

By definition, $h_1(x) = Q[x^{(0)}] + Q[256 + x^{(2)}]$. So the equalities $P[u]^{(0)} = P[v]^{(0)}$ and $P[u]^{(2)} = P[v]^{(2)}$ give $h_1(P[u]) = h_1(P[v])$ and this in turn gives $s_{u+12} \oplus s_{v+12} = P[u+12] \oplus P[v+12]$ by Lemma 2. □

The Glimpse Main Theorem [3, 4] is an important result on the weakness of RC4 stream cipher. It states that at any round, $Prob(S[j] = i - z) = Prob(S[i] = j - z) \approx 2^{-7}$, where S is the internal state of RC4, i and j are the deterministic and pseudo-random indices respectively and z is the keystream output byte. This result quantifies the leakage of state information into the keystream. Note that the leakage probability is twice the random association 2^{-8}. Our Theorem 6 is a Glimpse-like theorem on HC-128 that leaks state information into the keystream with a probability $\approx 2^{-15}$ which is much more than 2^{-31} (two times the random association 2^{-32}), and is in fact two times the square-root of the random association.

6 Conclusion

In this paper, we study the linear approximation of the feedback functions g_1, g_2 of HC-128. Using this result, we extend the least significant bitwise distinguisher proposed by the designer himself to all the bits (except one) of the 32-bit keystream output word. We also studied in detail the idea of Dunkelman towards secret state information leakage in keystream output words. Though our results do not have any immediate threat to the applicability of HC-128, these ideas identify weaknesses of the cipher that may provide further insight.

References

1. http://www.ecrypt.eu.org/stream/
2. O. Dunkelman. A small observation on HC-128. http://www.ecrypt.eu.org/stream/phorum/read.php?1,1143 Date: November 14, 2007.
3. R. J. Jenkins. ISAAC and RC4. 1996.
   Available at `http://burtleburtle.net/bob/rand/isaac.html` [last accessed on July 18, 2008].
4. I. Mantin. A Practical Attack on the Fixed RC4 in the WEP Mode. ASIACRYPT 2005, pages 395-411, volume 3788, Lecture Notes in Computer Science, Springer.
5. O. Staffelbach and W. Meier. Cryptographic Significance of the Carry for Ciphers Based on Integer Addition. CRYPTO 1990, pages 601-614, vol. 537, Lecture Notes in Computer Science, Springer.
6. H. Wu. The Stream Cipher HC-128. http://www.ecrypt.eu.org/stream/hcp3.html