A Linear Approximation to Addition of Three Integers and Its Implication to HC-128

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

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

Abstract. In this paper, we prove that the addition modulo 2^n of three n-bit integers has good linear approximation. We use this result 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.

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 [3] 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.

Let us first present a brief outline to the linear approximation of addition of n-bit integers modulo 2^n. Consider three integers $X = (X_{n-1}, \ldots, X_0), Y = (Y_{n-1}, \ldots, Y_0), Z = (Z_{n-1}, \ldots, Z_0)$ of n-bits each. Let $S = (X+Y) \bmod 2^n$, the addition modulo 2^n and $T = X \oplus Y$, the GF(2) addition corresponding to each bit. Similarly consider $S' = (X+Y+Z) \bmod 2^n$, and $T' = X \oplus Y \oplus Z$.

For $n = 8$, the probabilities of $S_i = T_i$ and $S'_i = T'_i$ are presented in the following table.

i	$Prob(S_i = T_i)$	$Prob(S'_i = T'_i)$
0	1.00000000	1.00000000
1	0.75000000	0.50000000
2	0.62500000	0.37500000
3	0.56250000	0.34375000
4	0.53125000	0.33593750
5	0.51562500	0.33398438
6	0.50781250	0.33349609
7	0.50390625	0.33337402

We like to point out that while $Prob(S_i = T_i)$ tends to $\frac{1}{2}$ as i increases, $Prob(S'_i = T'_i)$ tends to $\frac{1}{3}$. This shows that $Prob(S'_i = 1 \oplus X_i \oplus Y_i \oplus Z_i)$ is approximately $\frac{2}{3}$ for $i \geq 2$ and thus the i-th bit ($i \geq 2$) of addition modulo 2^n of three integers is highly correlated to the complement of the bitwise XOR of the integers. We theoretically prove this result in Section 3. The keystream output generation of HC-128 is actually of the form $S = (\alpha \oplus (\beta + ((\gamma \oplus \delta) + \tau))) \bmod 2^{32}$, where $S, \alpha, \beta, \gamma, \delta, \tau$ are 32-bit integers. One may have a look at the functions g_1, g_2 of HC-128 in this regard. Given our observation, $Prob(S_i = 1 \oplus T_i) \approx \frac{2}{3}$ for $i \geq 2$, where $T = \alpha \oplus \beta \oplus \gamma \oplus \delta \oplus \tau$.

In [3], 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". Our study, related to linear approximation of addition of three integers characterize the distinguisher for all other bits and we show that for each of the bits 2 to 31, the distinguisher is almost of the same strength as the distinguisher proposed for the least significant bit in [3]. We present this analysis in Section 4. To be precise, applying the linear approximation results in Section 3, we show that

- there are 30 many slightly weaker distinguishers other than the one described in [3] at bit level;
- 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 [3, 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 A Linear Approximation to Addition of Three Integers

In this section, we investigate the effectiveness of the approximation when modulo 2^n additions of three n-bit numbers are replaced by bitwise XOR's. This result can be used to approximate the feedback function in HC-128 and has implication towards the security of HC-128. Nonlinearity analysis of modulo addition of two integers have been studied in literature with great detail (one may refer to [5, 9, 6] and the references therein). However, these results do not cover our study related to addition of three integers. For two n-bit numbers, 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. Interestingly, this is not the case in the XOR-approximation of modulo addition of three numbers, as demonstrated in Theorem 1 at the end of this section.

Let X, Y, Z be three n-bit integers; $R = (X + Y + Z) \bmod 2^n$, $W = X \oplus Y \oplus Z$, the bitwise XOR. The i-th bit of the binary representation of X is written as X_i and similarly for other integers. We wish to compute $Prob(R_i = W_i)$ for $i \geq 0$. Here and also in the discussion below, probabilities are computed over independent and uniform random choices of the bits of X, Y and Z.

To do this, we need to consider the carry produced in the i-th step of the addition of X, Y and Z. Since, three bits are involved, the carry can take the values $0, 1$ and 2. The 2-bit carry produced in the i-th step will be denoted by (B_i, A_i) and we assume $A_{-1} = B_{-1} = 0$. So, $R_i = A_{i-1} \oplus X_i \oplus Y_i \oplus Z_i$ and hence $Prob(R_i = W_i) = Prob(A_{i-1} = 0)$. So, we need to analyse the sequence A_i.

Proposition 1. *For $i \geq 0$, we have $A_i = 1 \oplus B_{i-1} \oplus S_i$ and $B_i = b_{i-1}(1 \oplus S_i) \oplus A_{i-1}X_iY_iZ_i$, where*

$$S_i = \begin{cases} 1 \text{ if } wt(A_{i-1}, X_i, Y_i, Z_i) = 0, 1 \text{ or } 4; \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

Proof. Consider the i-th step of the addition. This looks as follows.

$$\begin{array}{cccc} & & & B_{i-1}\ A_{i-1} \\ & & & X_i \\ & & & Y_i \\ & & & Z_i \\ \hline B_i & & A_i & R_i \end{array}$$

Consider the addition of the first column (from the right). This produces a carry value of 0 into the second column if and only if the weight (number of 1's) of (A_{i-1}, X_i, Y_i, Z_i) is $0, 1$ or 4. From the definition of S_i, the value of this carry from the first into the second column is $1 \oplus S_i$. Now, A_i is 1 if and only if B_{i-1} and $(1 \oplus S_i)$ are equal. This gives the expression for A_i. A similar reasoning gives the expression for B_i. □

The following result is obtained from Proposition 1.

Proposition 2. *If $Prob(A_{i-1} = 0) = \delta_{i-1}$, then $Prob(S_i = 0) = \frac{1+\delta_{i-1}}{4}$.*

Proof. X_i, Y_i, Z_i are independent of A_{i-1}. So, we can write

$$\begin{aligned} Prob(S_i = 0) &= Prob(A_{i-1} = 0)Prob(wt(X_i, Y_i, Z_i) = 0 \text{ or } 1) \\ &\quad + Prob(A_{i-1} = 1)Prob(wt(X_i, Y_i, Z_i) = 0 \text{ or } 3) \\ &= \delta_{i-1} \times \frac{1}{2} + (1 - \delta_{i-1})\frac{1}{4} \\ &= \frac{1 + \delta_{i-1}}{4}. \end{aligned}$$

□

Lemma 1. *For $i \geq 0$,*

$$Prob(A_i = 0) = \frac{1}{2}\Gamma_{i-1}(0,0) + \frac{1}{4}\Gamma_{i-1}(0,1) + \frac{1}{2}\Gamma_{i-1}(1,0) + \frac{3}{4}\Gamma_{i-1}(1,1). \tag{2}$$

where for $i \geq -1$, and $\mu_1, \mu_2 \in \{0,1\}$, $\Gamma_i(\mu_1, \mu_2) \stackrel{\triangle}{=} Prob(B_i = \mu_1, A_i = \mu_2)$.

Note. $\Gamma_{-1}(0,0) = 1$ and $\Gamma_{-1}(0,1) = \Gamma_{-1}(1,0) = \Gamma_{-1}(1,1) = 0$.

Proof. We compute as follows.

$$\begin{aligned}
Prob(A_i = 0) &= Prob(B_{i-1} \neq S_i) \\
&= Prob(B_{i-1} = 0, S_i = 1) + Prob(B_{i-1} = 1, S_i = 0) \\
&= Prob(B_{i-1} = 0, A_{i-1} = 0, wt(X_i, Y_i, Z_i) = 0 \text{ or } 1) \\
&\quad + Prob(B_{i-1} = 0, A_{i-1} = 1, wt(X_i, Y_i, Z_i) = 0 \text{ or } 3) \\
&\quad + Prob(B_{i-1} = 1, A_{i-1} = 0, wt(X_i, Y_i, Z_i) = 2 \text{ or } 3) \\
&\quad + Prob(B_{i-1} = 1, A_{i-1} = 1, wt(X_i, Y_i, Z_i) = 1 \text{ or } 2) \\
&= \Gamma_{i-1}(0,0) Prob(wt(X_i, Y_i, Z_i) = 0 \text{ or } 1) \\
&\quad + \Gamma_{i-1}(0,1) Prob(wt(X_i, Y_i, Z_i) = 0 \text{ or } 3) \\
&\quad + \Gamma_{i-1}(1,0) Prob(wt(X_i, Y_i, Z_i) = 2 \text{ or } 3) \\
&\quad + \Gamma_{i-1}(1,1) Prob(wt(X_i, Y_i, Z_i) = 1 \text{ or } 2).
\end{aligned}$$

Since X_i, Y_i and Z_i are independent and uniformly distributed over $\{0,1\}$,
$Prob(wt(X_i, Y_i, Z_i) = 0 \text{ or } 1) = 1/2$. Similarly, we obtain the other values. □

The next task is to obtain a recurrence for $\Gamma_i(\mu_1, \mu_2)$.

Lemma 2. *For $i \geq 0$, the following holds.*

$$\left.\begin{aligned}
\Gamma_i(0,0) &= \tfrac{1}{2}\Gamma_{i-1}(0,0) + \tfrac{1}{8}\Gamma_{i-1}(0,1) \\
\Gamma_i(0,1) &= \tfrac{1}{2}\Gamma_{i-1}(0,0) + \tfrac{3}{4}\Gamma_{i-1}(0,1) + \tfrac{1}{2}\Gamma_{i-1}(1,0) + \tfrac{1}{8}\Gamma_{i-1}(1,1) \\
\Gamma_i(1,0) &= \tfrac{1}{8}\Gamma_{i-1}(0,1) + \tfrac{1}{2}\Gamma_{i-1}(1,0) + \tfrac{3}{4}\Gamma_{i-1}(1,1) \\
\Gamma_i(1,1) &= \tfrac{1}{8}\Gamma_{i-1}(1,1).
\end{aligned}\right\} \tag{3}$$

Proof. The proofs are similar and we only prove the second point. The event $(B_i = 0, A_i = 1)$ can be divided into four mutually disjoint events:

$$\begin{aligned}
&B_{i-1} = 0, A_{i-1} = 0, wt(X_i, Y_i, Z_i) = 2 \text{ or } 3; \\
&B_{i-1} = 0, A_{i-1} = 1, wt(X_i, Y_i, Z_i) = 1 \text{ or } 2; \\
&B_{i-1} = 1, A_{i-1} = 0, wt(X_i, Y_i, Z_i) = 0 \text{ or } 1; \\
&B_{i-1} = 1, A_{i-1} = 1, wt(X_i, Y_i, Z_i) = 1.
\end{aligned}$$

Consequently, $\Gamma_i(0,1) = Prob(B_i = 0, A_i = 1)$ is the sum of the probabilities of the above four events. Using the independence of X_i, Y_i, Z_i from A_{i-1} and B_{i-1}, each probability becomes the product of two probabilities. For example,

$Prob(B_{i-1} = 0, A_{i-1} = 0, wt(X_i, Y_i, Z_i) = 2 \text{ or } 3)$
$= \Gamma_{i-1}(0,0) Prob(wt(X_i, Y_i, Z_i) = 2 \text{ or } 3) = \frac{1}{2}\Gamma_{i-1}(0,0)$ and similarly for the other three cases. This gives the required expression for $\Gamma_i(0,1)$. □

Equation (2) expresses $Prob(A_i = 0)$ in terms of the four Γ_{i-1}'s. On the other hand, (3) expresses each Γ_i in terms of the four Γ_{i-1}'s. If we put these two together, then we can express $Prob(A_i = 0)$ in terms of the four Γ_{i-2}'s. Clearly this process can be repeated, so that we can express $Prob(A_i = 0)$ in terms of the four Γ_{i-j-1}'s for $j = 0, 1, \ldots, i$. The following result makes this precise.

Lemma 3. *For $i \geq 0$,*

$$Prob(A_i = 0) = \delta_j^{(0)} \Gamma_{i-j-1}(0,0) + \delta_j^{(1)} \Gamma_{i-j-1}(0,1) + \delta_j^{(2)} \Gamma_{i-j-1}(1,0) + \delta_j^{(3)} \Gamma_{i-j-1}(1,1), \quad (4)$$

where $\delta_0^{(0)} = 1/2$, $\delta_0^{(1)} = 1/4$, $\delta_0^{(2)} = 1/2$, $\delta_0^{(3)} = 3/4$ and for $0 \leq j \leq i-1$,

$$\left.\begin{array}{l} \delta_{j+1}^{(0)} = \frac{1}{2}\delta_j^{(0)} + \frac{1}{2}\delta_j^{(1)} \\ \delta_{j+1}^{(1)} = \frac{1}{8}\delta_j^{(0)} + \frac{3}{4}\delta_j^{(1)} + \frac{1}{8}\delta_j^{(2)} \\ \delta_{j+1}^{(2)} = \frac{1}{2}\delta_j^{(1)} + \frac{1}{2}\delta_j^{(2)} \\ \delta_{j+1}^{(3)} = \frac{1}{8}\delta_j^{(1)} + \frac{3}{4}\delta_j^{(2)} + \frac{1}{8}\delta_j^{(3)}. \end{array}\right\} \quad (5)$$

Proof. The case of $j = 0$ is exactly Lemma 1 and the recurrence (5) follows by induction using Lemma 2. □

This leads to the following theorem.

Theorem 1. *For $i \geq -1$, $Prob(A_i = 0) = \dfrac{1}{3}\left(1 + \dfrac{1}{2^{2i+1}}\right)$.*

Proof. For $i = -1$, we have $Prob(A_i = 0) = 1$ and this verifies with the given expression.

For $i \geq 0$, from Lemma 3, we can write

$$\begin{aligned} Prob(A_i = 0) &= \delta_i^{(0)} \Gamma_{-1}(0,0) + \delta_i^{(1)} \Gamma_{-1}(0,1) + \delta_i^{(2)} \Gamma_{-1}(1,0) + \delta_i^{(3)} \Gamma_{-1}(1,1) \\ &= \delta_i^{(0)}. \end{aligned}$$

The last equation holds since $\Gamma_{-1}(0,0) = 1$ and $\Gamma_{-1}(0,1) = \Gamma_{-1}(1,0) = \Gamma_{-1}(1,1) = 0$.

So, we are reduced to computing $\delta_i^{(0)}$. From (5), we see that $\delta_j^{(3)}$ does not influence $\delta_{j+1}^{(0)}, \delta_{j+1}^{(1)}$ and $\delta_{j+1}^{(2)}$. So, the first three equations of (5) can be solved independent of the fourth equation. It is easily verified by induction that the following constitutes the solution to these three equations.

$$\begin{aligned} \delta_j^{(0)} &= \frac{1}{3}\left(1 + \frac{1}{2^{2j+1}}\right); \\ \delta_j^{(1)} &= \frac{1}{3}\left(1 - \frac{1}{2^{2j+1}}\right); \\ \delta_j^{(2)} &= \frac{1}{3}\left(1 + \frac{1}{2^{2j+1}}\right). \end{aligned}$$

This gives the required result. □

4 Implication to HC-128

Here we present how the result of the previous section can be used in analysing HC-128. 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, $0 \le b \le n-1$, 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 numbers with $\mathcal{S} = (X_1 + X_2 + X_3) \bmod 2^n$. Then, for $0 \le b \le n-1$,*

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

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

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

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 found good linear approximation in this case. We have described in the previous section that the addition of three integers does not provide as good nonlinearity as the addition of two integers.

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 $0 \le b \le n-1$, 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 $0 \le b \le n-1$, we have

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

Following the same notation as in [3, 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.

Proposition 3. *The expected number of bits where two 32-bit integers s_i, ψ_i match is 21.5.*

Proof. The result follows 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 that the expected number of matches between s_i, ψ_i is $30 \cdot \frac{2}{3} + \frac{1}{2} + 1 = 21.5$. □

Proposition 3 shows the association of the HC-128 keystream words s_i with its linear approximation ψ_i.

4.1 Extending the Distinguisher of [3] to Other Bits

In [3, Section 4], it was shown that 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 [3, Section 4], this distinguisher requires 2^{164} pairs of above keystream word combinations for a success probability 0.9772. It has been commented in [3] 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 here that the distinguisher works for all the bits (except one) in the keystream words. 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 [3, Section 4] by applying Corollary 1. The keystream output word of HC-128 is generated as $s_i = h_1(P[i \boxminus 12]) \oplus P[i]$, $0 \leq i \bmod 1024 < 512$. 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 $10 \leq i \bmod 1024 < 511$,

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

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.

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

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

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

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 $2 \leq b \leq 31$. In short, we can write, for $0 \leq b \leq 31$,

$$Prob(\Psi_i^b = H_i^b) = p_b,$$

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

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

Obviously, for $0 \leq b \leq 31$, $Prob(\Psi_i^b = H_i^b \oplus 1) = 1 - p_b$.

Thus, we can state the following technical result.

Lemma 4. *For* $1024\tau + 10 \leq j < i < 1024\tau + 511$ *and* $0 \leq b \leq 31$,

$$Prob(\Psi_i^b \oplus \Psi_j^b = H_i^b \oplus H_j^b) = q_b$$

where

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

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

Obviously, for $0 \leq b \leq 31$, $Prob(\Psi_i^b \oplus \Psi_j^b = H_i^b \oplus H_j^b \oplus 1) = 1 - q_b$.

Also we have the following result about the collision in $H(.)$.

Proposition 4. *[3, Theorem 1] For* $1024\tau + 10 \leq j < i < 1024\tau + 511$ *and* $0 \leq b \leq 31$,

$$Prob(H_i^b = H_j^b) = \frac{1}{2} + 2^{-81}.$$

Obviously, $Prob(H_i^b = H_j^b \oplus 1) = \frac{1}{2} - 2^{-81}$.

Combining the above results, we get the following theorem.

Theorem 2. *For* $1024\tau + 10 \leq j < i < 1024\tau + 511$, $Prob(\Psi_i^b = \Psi_j^b) = \rho_b$, *where*

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

Proof. $Prob(\Psi_i^b = \Psi_j^b)$
$= Prob(\Psi_i^b \oplus \Psi_j^b = H_i^b \oplus H_j^b) \cdot Prob(H_i^b = H_j^b) + Prob(\Psi_i^b \oplus \Psi_j^b = H_i^b \oplus H_j^b \oplus 1) \cdot Prob(H_i^b = H_j^b \oplus 1)$.
Substituting values from Lemma 4 and Proposition 4, 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 [3, 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 numbers, 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 3. *Let M denotes the number of matches between the corresponding bits of Ψ_i and Ψ_j, for $1024\tau + 10 \leq j < i < 1024\tau + 511$. Then the expected number of matches is given by $E(M) = 16 + \frac{13}{12} \cdot 2^{-79}$.*

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 2, 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 2, 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 [3] at bit level (Theorem 2);
- further, all these distinguishers can be taken together to mount a word level distinguisher for HC-128 (Theorem 3).

These distinguishers have not been identified in [3].

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 5 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 5. *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 6, 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 6. *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 6 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 2. *For $0 \le t \le n$, 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 6, 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 6) 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 4. *In HC-128, consider a block of 512 many keystream words corresponding to array P. For $0 \le u \neq v \le 511$, $Prob\big((s_u \oplus s_v) = (P[u] \oplus P[v])\big) = \alpha_{8,32} > 2^{-16}$.*

Proof. The result follows from Lemma 5 and Lemma 6. □

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

Theorem 5. *In HC-128, consider a block of 512 many keystream words corresponding to array P. For any u, v, $0 \le u \neq v < 500$, 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 5, $s_u^{(b)} \oplus s_v^{(b)} = P[u]^{(b)} \oplus P[v]^{(b)}$ 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,

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

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 5. □

The Glimpse Main Theorem [4, 7] 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 5 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 addition of three integers and found that the addition of three integers does not provide good nonlinearity. This result is used for analysis of HC-128 and 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. H. Wu. The Stream Cipher HC-128. http://www.ecrypt.eu.org/stream/hcp3.html
4. R. J. Jenkins. ISAAC and RC4. 1996.
   Available at `http://burtleburtle.net/bob/rand/isaac.html` [last accessed on July 18, 2008].
5. H. Lipmaa and S. Moriai. Efficient Algorithms for Computing Differential Properties of Addition. FSE 2001, pages 336-350, vol. 2355, Lecture Notes in Computer Science, Springer.
6. H. Lipmaa, J. Wallen and P. Dumas. On the Additive Differential Probability of Exclusive-Or. FSE 2004, pages 317-331, vol. 3017, Lecture Notes in Computer Science, Springer.
7. 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.
8. I. Mantin and A. Shamir. A Practical Attack on Broadcast RC4. FSE 2001, pages 152-164, vol. 2355, Lecture Notes in Computer Science, Springer.
9. J. Wallén. Linear Approximations of Addition Modulo 2^n. FSE 2003, pages 261-273, vol. 2887, Lecture Notes in Computer Science, Springer.