A New Improved Distinguisher for HC-128

Subhabrata Sen[1], Rudradev Sengupta[1], Subhamoy Maitra[1], Goutam Paul[2],
Shashwat Raizada[1]

[1] Indian Statistical Institute,
203 B T Road, Kolkata 700 108, India.
subhabratasen.19@gmail.com, rudradevsengupta@gmail.com,
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 present a new distinguisher for HC-128 which is the best known so far. The distinguisher requires approximately 2^{106} keystream words with success probability 0.9772.

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

1 Introduction

The eSTREAM [2] 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 [3] in the eSTREAM discussion forum to show that the keystream words of HC-128 leak information regarding secret states. Recently, generalization of these results has been studied in [4]. In this paper, we identify a new and improved distinguisher for HC-128. To the best of our knowledge, this is currently the strongest distinguisher available.

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. In [4], the distinguisher is extended for other bits too. The distinguishers presented in [6, 4] require approximately 2^{156} words of keystream with success probability of 0.9772. Our distinguisher requires approximately 2^{106} keystream words with same success probability. For every block of 512 many keystream words of HC-128, corresponding to either the P or Q array, we show that XOR of the least significant bits (LSBs) of four keystream words (two taken from the upper sub-block of 256 keystream words and two taken from the lower sub-block of 256 keystream words) is biased towards 0. The distinguisher can also be extended for bits other than the LSBs.

The complete study of the new distinguisher is presented in Section 3. Let us 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:

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

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

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.

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

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

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

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

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 Our New Distinguisher

In this section we present our new distinguisher. Before getting into the technical details, let us explain one more notation. For any n-bit integer I, $[I]^b$ denotes the b-th least significant bit, $0 \leq b \leq n - 1$. Thus $[I]^0$ denotes the LSB of I. Also in the following discussion, we abuse the notation of s_i described in Section 2.3. Here, by s_u, we mean a keystream word generated in the u-th step in a block of 512 words corresponding to either the P array (completely) or the Q array (completely). As we will be concentrating on the relationship between four keystream words in a block of 512, (to keep the notation simple) we do not use any index to identify different blocks.

Consider the term $h_2(Q[\alpha])$, for $0 \leq \alpha \leq 511$. Note that

$$h_2(Q[\alpha]) = P[\beta^{(0)}] + P[256 + \beta^{(2)}],$$

where $\beta = Q[\alpha]$ and then we get

$$h_2(Q[\alpha]) = (s_{\beta^{(0)}} \oplus h_1(P[\beta^{(0)} \boxminus 12])) + (s_{256+\beta^{(2)}} \oplus h_1(P[256 + \beta^{(2)} \boxminus 12])). \quad (1)$$

As noted in [6], for the least significant bit, '+' can be replaced by '$\oplus$' and hence,

$$[s_{\beta^{(0)}} \oplus s_{256+\beta^{(2)}}]^0 = [h_1(P[\beta^{(0)} \boxminus 12]) \oplus h_1(P[256 + \beta^{(2)} \boxminus 12]) \oplus h_2(\beta)]^0. \quad (2)$$

Denoting $u = \beta^{(0)}, l = 256 + \beta^{(2)}, \mu = P[\beta^{(0)} \boxminus 12], \nu = P[256 + \beta^{(2)} \boxminus 12]$, we have

$$[s_u \oplus s_l]^0 = [h_1(\mu) \oplus h_1(\nu) \oplus h_2(\beta)]^0.$$

Thus for $u \neq u'$ and $l \neq l'$, we get,

$$[s_u \oplus s_l]^0 = [s_{u'} \oplus s_{l'}]^0,$$

if and only if

$$[h_1(\mu) \oplus h_1(\nu) \oplus h_2(\beta)]^0 = [h_1(\mu') \oplus h_1(\nu') \oplus h_2(\beta')]^0,$$

where μ', ν' and β' correspond to the indices u' and l' in the same manner as μ, ν and β correspond to the indices u and l.

We start concentrating on $Q[\alpha]$ by noting the expression of $h_2(Q[\alpha])$. This leads to the relationship among the keystream words generated corresponding to the previous P array.

The function h_1 uses actually 16 bits only from the 32-bit input. The situation is similar for h_2. Thus, $[h_1(.) \oplus h_1(.) \oplus h_2(.)]^0$ can be approximated (similar to the idea of [6, Section 4]) as a 48-bit-to-1-bit S-box. Thus, $Pr([h_1(\mu) \oplus h_1(\nu) \oplus h_2(\beta)]^0 = [h_1(\mu') \oplus h_1(\nu') \oplus h_2(\beta')]^0)$ is equal to the collision probability of a 48-bit-to-1-bit S-box. According to [6, Theorem 1], this is equal to $2^{-48} + 2^{-1} - 2^{-48-1}$. This immediately leads to the following result.

Lemma 1. *Consider the consecutive P and Q arrays in the execution of HC-128 and let s_t, $0 \leq t \leq 511$, be the keystream words generated corresponding to the P array. For $0 \leq \alpha \neq \alpha' \leq 511$, we have*

$$Pr\left([s_u \oplus s_l]^0 = [s_{u'} \oplus s_{l'}]^0\right) = \frac{1}{2} + \frac{1}{2^{49}},$$

where $u = Q[\alpha]^{(0)}$, $l = 256 + Q[\alpha]^{(2)}$, $u' = Q[\alpha']^{(0)}$ and $l' = 256 + Q[\alpha']^{(2)}$.

In a similar direction, one can start with $P[\alpha]$ by noting the expression of $h_1(P[\alpha])$. This will lead to similar relationship among the keystream words generated corresponding to the previous Q array.

Lemma 2. *Consider the consecutive Q and P arrays in the execution of HC-128 and let s_t, $0 \leq t \leq 511$, be the keystream words generated corresponding to the Q array. For $0 \leq \alpha \neq \alpha' \leq 511$, we have*

$$Pr\left([s_u \oplus s_l]^0 = [s_{u'} \oplus s_{l'}]^0\right) = \frac{1}{2} + \frac{1}{2^{49}},$$

where $u = P[\alpha]^{(0)}$, $l = 256 + P[\alpha]^{(2)}$, $u' = P[\alpha']^{(0)}$ and $l' = 256 + P[\alpha']^{(2)}$.

Given α, α', there is no way to observe the values of $Q[\alpha], Q[\alpha']$ (or $P[\alpha], P[\alpha']$) and hence we cannot identify the indices u, l, u', l'.

We overcome this problem in the following manner. We know that any element $Q[\alpha]$ (or $P[\alpha]$), $0 \le \alpha \le 511$ provides one pair of keystream words of the form (s_u, s_l). So there are $\binom{512}{2}$ many quadruples of the form (u, l, u', l') for which Lemma 1 (or Lemma 2) holds. We refer to these quadruples as *favourable quadruples*. The following result uses Lemma 1 and Lemma 2 to compute the expression of the probability for "any" quadruple (u, l, u', l') where the pair (u, u') corresponds to the initial half $(0 \le u \ne u' \le 255)$ and the pair (l, l') corresponds to the later half $(256 \le l \ne l' \le 511)$ of the array Q (or P). Now we present our main result that will be used to find the new distinguisher.

Theorem 1. *Let s_t, $0 \le t \le 511$, be the keystream words generated corresponding to either P or Q array. For $0 \le u \ne u' \le 255$, $256 \le l \ne l' \le 511$,*

$$Pr\left([s_u \oplus s_l]^0 = [s_{u'} \oplus s_{l'}]^0\right) \approx \frac{1}{2} + \frac{1}{2^{62}}.$$

Proof. From the ranges of u, u', l, l', it is clear that there are $\binom{256}{2}^2$ many quadruples of the form (u, l, u', l'). Let F be the event that an arbitrary quadruple (u, u', l, l') is favourable and further let E be the event $\left([s_u \oplus s_l]^0 = [s_{u'} \oplus s_{l'}]^0\right)$.

We can choose any α, α' for $0 \le \alpha \ne \alpha' \le 511$ to build one equation of the form $[s_u \oplus s_l]^0 = [s_{u'} \oplus s_{l'}]^0$ with the following constraint. If at least one of the equalities $(Q[\alpha]^{(0)} = Q[\alpha']^{(0)})$ or $(Q[\alpha]^{(2)} = Q[\alpha']^{(2)})$ hold(s), then we cannot form the above combination of keystream bits to generate the equation. The situation is similar for the case of P. However, the expected number of such cases is very small (around 4 out of $\binom{512}{2} \approx 2^{17}$) if we consider that Q (or P) contains 512 many 32-bit integers chosen uniformly at random from the set of all 32-bit integers. Thus, we can approximate $Pr(F) = \frac{\binom{512}{2}}{\binom{256}{2}^2}$.

Further, from Lemma 1, Lemma 2, $Pr(E|F) = \frac{1}{2} + \frac{1}{2^{49}}$. We can assume that for a non-favourable quadruple, the event E occurs due to random association only, i.e., $Pr(E|F^C) = \frac{1}{2}$, where F^C is the complement of the event F. Thus, $Pr(E) = Pr(F) \cdot Pr(E|F) + Pr(F^C) \cdot Pr(E|F^C)$
$= \frac{\binom{512}{2}}{\binom{256}{2}^2} \cdot (\frac{1}{2} + \frac{1}{2^{49}}) + \left(1 - \frac{\binom{512}{2}}{\binom{256}{2}^2}\right) \cdot \frac{1}{2} \approx \frac{1}{2} + \frac{1}{2^{62}}$. □

Hence, Theorem 1 gives us a distinguisher. The number of keystream words required to mount the above distinguisher is computed in Theorem 2 below.

Theorem 2. *HC-128 can be distinguished from an ideal random word generator by observing 2^{106} keystream words with a success probability of 0.9772.*

Proof. According to Theorem 1, the event $\left([s_u \oplus s_l]^0 = [s_{u'} \oplus s_{l'}]^0\right)$ based on which the distinguisher is constructed occurs with a probability $p(1 + q)$, where $p = \frac{1}{2}$ and $q \approx \frac{1}{2^{61}}$. According to [1, Section 4.1], to get a success probability of 0.9772, one would require $\frac{4^2}{pq^2} = 2^{127}$ many samples. In our case, each sample consists of a set of 4 keystream words of the form $(s_u, s_{u'}, s_l, s_{l'})$. Since each block of $512 = 2^9$ many keystream words (corresponding to either the array P or the array Q) gives $\binom{256}{2}^2 \approx 2^{30}$ many samples, one needs $2^{127+9-30} = 2^{106}$ many keystream words to mount the above distinguisher. □

In terms of data complexity, we have a significant improvement over [6, 4], where the number of keystream words required is around 2^{156} for the same success probability.

So far we have concentrated on the LSBs and now we extend this to other bits also. We have replaced the '+' in Equation 1 to get Equation 2 relating the least significant bits. Note that Equation 2 holds with probability 1. However, one may write similar equations for the other bits using the result [5] that for two n-bit integers X and Y, if $S = X + Y$ and $T = X \oplus Y$, then $Pr(S^b = T^b) = \frac{1}{2}(1 + \frac{1}{2^b})$. This XOR-approximation of sum can be used to extend the least significant bit based distinguisher (Theorem 1) to the other bits too. As the analysis is similar, we present the following result without proof.

Theorem 3. *Let s_t, $0 \leq t \leq 511$, be the keystream words generated corresponding to either P or Q array. For $0 \leq u \neq u' \leq 255$, $256 \leq l \neq l' \leq 511$,*

$$Pr\left([s_u \oplus s_l]^b = [s_{u'} \oplus s_{l'}]^b\right) \approx \frac{1}{2} + \frac{1}{2^{62+2b}},$$

for $0 \leq b \leq 31$. Thus, if one concentrates on the b-th bit of each keystream word, HC-128 can be distinguished from an ideal random word generator by observing 2^{106+4b} keystream words with a success probability of 0.9772.

Based on Theorem 3, we can find the following result.

Theorem 4. *Let s_t, $0 \leq t \leq 511$, be the keystream words generated corresponding to either P or Q array. Then the expected number of 0's in the bit pattern of $s_u \oplus s_l \oplus s_{u'} \oplus s_{l'}$ is $16 + \frac{1}{3}(\frac{1}{2^{60}} - \frac{1}{2^{124}})$, where $0 \leq u \neq u' \leq 255$, $256 \leq l \neq l' \leq 511$.*

Proof. Let $\psi = s_u \oplus s_l \oplus s_{u'} \oplus s_{l'}$. Let $m_b = 1$, if $[\psi]^b = 0$; otherwise, let $m_b = 0$, $0 \leq b \leq 31$. Hence, the total number of zeros in the bit pattern of ψ is given by $M = \sum_{b=0}^{31} m_b$. From Theorem 3, we have $Prob(m_b = 1) = \frac{1}{2} + \frac{1}{2^{62+2b}}$. Hence, $E(m_b) = \frac{1}{2} + \frac{1}{2^{62+2b}}$ and by linearity of expectation, $E(M) = \sum_{b=0}^{31} E(m_b) = 16 + \frac{1}{3}(\frac{1}{2^{60}} - \frac{1}{2^{124}})$. □

Our result is much sharper than [4, Theorem 4], where the expected number of 0's in the bit pattern of the XOR of 10 properly chosen different keystream words was $16 + \frac{13}{12} \cdot 2^{-79}$.

4 Conclusion

In this paper we present the currently best known distinguisher for HC-128 that requires 2^{106} keystream words for a success probability of 0.9772. This distinguisher involves LSBs of four properly chosen keystream words. Our distinguisher can be extended to any bit of the keystream words also.

References

1. R. Basu, S. Ganguly, S. Maitra and G. Paul. A Complete Characterization of the Evolution of RC4 Pseudo Random Generation Algorithm. *Journal of Mathematical Cryptology*, pages 257-289, vol. 2, no. 3, October, 2008.
2. http://www.ecrypt.eu.org/stream/ [last accessed on April 15, 2009].
3. O. Dunkelman. A small observation on HC-128. A message dated November 14, 2007 is available at http://www.ecrypt.eu.org/stream/phorum/read.php?1,1143 [last accessed on April 15, 2009].
4. S. Maitra, G. Paul and S. Raizada. Some Observations on HC-128. Pages 527–539 in the workshop pre-proceedings, WCC 2009.
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 [last accessed on April 15, 2009].