Weaknesses in HENKOS Stream Cipher

Prasanth Kumar Thandra and S.A.V. Satya Murty
Indira Gandhi Centre for Atomic Research,
Kalpakkam, Tamilnadu 603102.

R Balasubramanian,
Institute of Mathematical Sciences,
Chennai.

prasanth@igcar.gov.in, satya@igcar.gov.in. balu@imsc.res.in.

Abstract

HENKOS[1] is a synchronous stream cipher posted by Marius Oliver Gheorghita to eprint. In this paper we are going to present some weaknesses in the cipher. We first present a chosen IV attack which is very straight forward attack on the cipher. Second we present a group of weak keys.

Key words: synchronous stream ciphers, weak keys/ related keys, chosen IV attack.

1. Introduction and Algorithm description:

Synchronous stream cipher HENKOS is proposed by Marius Oliver Gheorghit to eprint for analyzing the algorithm by cryptologic community. At the first look of HENKOS, it can be observed that a 256 byte secret key is an undesirable quality of the algorithm. Also, the algorithm is not efficient in case of streaming data that has to encrypt instantaneously, this is because the algorithm generates a 256 bytes of key stream in each round.

In [1] author claimed that the algorithm is safe from chosen IV attacks and weak keys/ related keys. Till now no attacks are posted against the algorithm. In the present paper we show that the algorithm HENKOS is not safe against chosen IV attacks and we found that there are many weak keys. The attacks we proposed are very straight forward and very useful for the young attackers to understand the chosen IV and weak key attacks clearly. The description of the algorithm is given below.

HENKOS :

The algorithm contains three main components that are 256 bytes in length. A secret key of 256 bytes acts as an array called master key, MK. Another array, called master key transformed (MKT), of length 256 bytes is initialized by MK as follows

$$MKT_i = \text{mirror}\ [(\textstyle\sum_{j=0..i} MK(j)\ )\%256)]\%256, \quad \text{for}\ \ i = 0..255; \qquad \text{--------- (1)}$$

Where, mirror is an operation which acts as in the next example. E.g. Mirror image of the number 255 is 552.

After the master key transformation initialization vector, IV, of length 256 bytes will be considered as another array called data key, DK. The data key is initialized by Data-Key-

Initialization. The Data-Key-Initialization comprises 64 rounds of recursive operations of two functions t (or SW called switch function) and f (or AD called additive function).

The "switch" function (SW), which will mix the bytes of the data key as follows:
The byte j is switched with byte k in the data key, where j is the number from the MK in the i position and k is the number from the MKT in the i position.

$$\left.\begin{array}{l} DK_j = DK_j \text{ XOR } DK_k ; \\ DK_k = DK_j \text{ XOR } DK_k ; \\ DK_j = DK_j \text{ XOR } DK_k : \\ \text{Where, } j = MK_i \text{ and } k=MKT_i;\ i = 0..255; \end{array}\right\} \quad \text{--------- (2)}$$

The additive function, AD, will replace the byte from each position with the sum between it and the byte from the right, except the last byte who is added with first byte.

$$\left.\begin{array}{l} DK_i = DK_i + DK_{i+1} \text{ modulo } 256 \qquad i = 0.....254; \\ \\ DK_{255} = DK_{255} + DK_0; \end{array}\right\} \quad \text{---------- (3)}$$

DK_0=initial value, not calculated in this cycle. After these two transformations, obtain an intermediate data key; to initialize the data key properly, these cycles will be repeated 64 times, without producing any output (in the figure the initialization key is the dashed-line box). After the last cycle a DK is released. This is the first internal state S_0. Then the output key stream, of length 256 bytes, is calculated as follows

$$Z_i=S_i \text{ XOR } t(S_i) ; \quad \text{---------- (4)}$$

For generation of a keystream with predefined length, function g must be applied as long as necessary, using the functions described above in the data key initialization cycle.

Encryption/decryption: The encryption/decryption between the plaintext/ciphertext is done using XOR:

ci = mi XOR zi ; mi= ci XOR zi ;

Where, ci = ciphertext, mi = plaintext, zi = keystream;

2. A Chosen IV attack (weakness) on HENKOS:

In a chosen IV attack, attacker uses IV(s) of his choice and tries to achieve the knowledge over either the plaintext or the bits of secret key by analyzing the ciphertext. In case of a known plaintext attacker try find out the correlations in the output keystreams of different IVs to get the knowledge over secret key. When the key and plaintext both are secret attacker uses IVs of his choice to find the plaintext bits. The present attack on HENKOS

belongs to the second category where both key and plaintext are unknown to the attacker and the attacker is provided with ciphertexts and choice of choosing IV(s).
The output keystream of HENKOS is generated by $Z_i=S_i$ XOR $t(S_i)$, where the internal states values are calculated from DK , which is initialized by IV. A chosen IV attack can be mounted on HENKOS by using the weakness in the Data-Key-Initialization process and the output function.

Choose an IV of 256 bytes with all the values of individual bytes equal to zero. So, for this IV/DK, it can be followed from eq(2) and eq(3) the value of all the individual bytes t(DK), f(t(DK)) (say f(t(x)))=h(x)) and h^{64}(DK) and so on will be always zero irrespective of the choice of MK. This directly leads to the output keystream values to be always zero. This is true for the output keystream at any round of the stream cipher HENKOS. Hence, in this case both plaintext and ciphertext are same. This shows a clear weakness in the IV setup and updating of internal state of the cipher HENKOS. Fig 1 shows the arrays DK, t(DK), h(DK) and h^{64}(DK) for an IV with all its individual byte values zero.

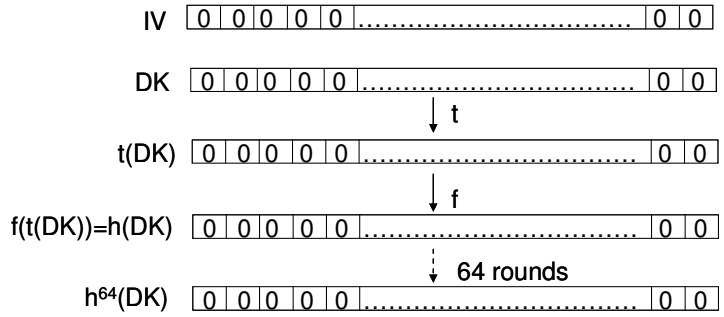

FIG.1 The values of DK, t(DK),f(t(DK))…..for IV=0.

3. Weak Keys of HENKOS

We found many weak keys for the HENKOS stream ciphers. Weak keys, we mean here, can be expressed as follows

(a) A set-of-keys which gives same output with a particular IV or set of IVs.
(b) A particular key which gives same output with a set-of-IVs.
(c) a particular key, IV pair gives the same output as another key, IV pair

It can be noticed that the weakness of HENKOS mentioned in section 2 comes under he category (a), where the specific IV gives same output with any key. Now here we are going to present a set of weak keys which comes under the category (a). That is all members of the set-of-keys will give the same output keystream with a particular IV. The members of this set-of-keys are chosen in such a way that they can exploit the output function (4). From (4) it can be clearly observed that Z_i (S_i XOR $t(S_i)$) will be always zero, independent of the choice of IV, if the t function does not change the internal state S_i. We observed different groups of keys which makes the t function ineffective (i.e. t(x) = x).

The members of first group-of-keys, MKs, which leads t(x) = x, can be described as follows. All the 128 values at the even indices of each MK are set to zero. Out of the remaining 128 odd indices of MK, 127 values are set to zero. The value at the remaining odd index is L, where L is a member of the set called Symmetry. In this way the total number of keys in first group is 3201(i.e. 26*128-127).

$$\begin{aligned}\text{Symmetry} = \{&000,010,020,030,040,050,060,070,080,090\\ &101,111,121,131,141,151,161,171,181,191,\\ &202, 212, 222, 232, 242, 252\}\end{aligned} \qquad \text{---------- (5)}$$

If an MK which belongs to the above group is represented as $MK^{(L,M)}$ where M is the odd index of array $MK^{(L,M)}$ where the value L is. As an example if we consider L=222 and M=253. Now $MK^{(222,253)}$, $MKT^{(222,253)}$ are represented below in fig 2.

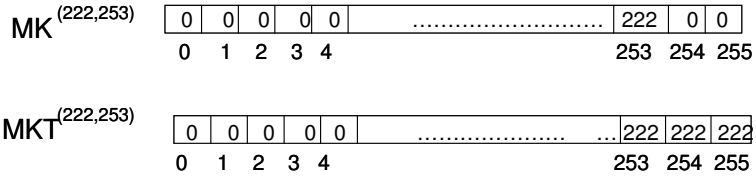

Fig.2 Arrays $MK^{(222,253)}$ and $MKT^{(222,253)}$

Now if eq(2), t function, is performed on any S_i with the above $MK^{(222,253)}$, $MKT^{(222,253)}$ pair the first 254 rounds of t function does not alter S_i at 255^{th} round $S_i[0]$ and $S_i[222]$ are exchanged and finally at 256^{th} round the contents of $S_i[0]$ and $S_i[222]$ are again exchanged. This leads the S_i to be unvaried even after t function, which intern leads the output keystream to be zero at any round and with any IV. This is true with any member of the group of keys. Hence for this group of keys

$$t(S_i)=S_i \text{ then } Z_i=S_i \text{ XOR } t(S_i) = S_i \text{ XOR } S_i = 0.$$

So, the total output keystream is same, zero, for all the members of the group irrespective of value of DK or IV. It can be noticed here that a particular member of the group always generates the zero output keystream for any IV. This comes under the category (b) of weak keys as mentioned.

A member of a second group-of-keys which does the same task of making t function ineffective is shown below in fig 3.

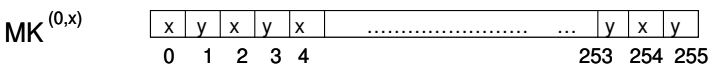

Fig 3: Array $MK^{(0, x)}$

Where, $MK^{(I, x)}$ represents an array with the follow properties

- "I" is an even value from the set {0, 4, 8, 12, 16, 20 …….244, 248, 252}
- All the elements at indexes lesser than "I" are set to zero.
- All the elements at even indexes which are greater than or equal to I are set to x.
- All the elements at odd indexes which are greater than I are set to y.

Where, x belongs to {1,2,3,4,5,…255} and value of y corresponding to an x is y=256-x. $MKT^{(0, x)}$ along with $MK^{(0, x)}$ is shown below fig 4.

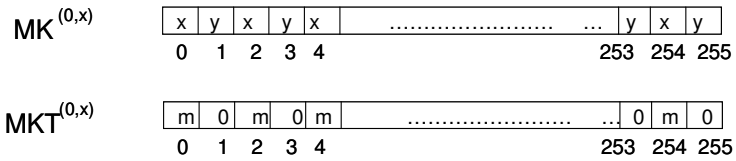

Fig 4: Arrays $MK^{(0, x)}$ and $MKT^{(0, x)}$

From this it can be directly observed that when a "t" function is applied on S_i which uses a member $MK^{(I, x)}$ of the second group, the contents of $S_i[x]$ and $S_i[m]$ are exchanged even number of times and also the contents $S_i[y]$ and $S_i[0]$ are exchanged even number of times which leaves S_i unchanged finally after complete application of "t" function. So, $t(S_i)=S_i$ hence,

$$Z_i=S_i \text{ XOR } t(S_i) = S_i \text{ XOR } S_i = 0.$$

There are many other group of weak keys which leaves $t(S_i)=S_i$. Below we have given, in fig 5, a member of such a group. The members, $MK^{(J, x, y)}$, of this group look similar to that of in fig 4.

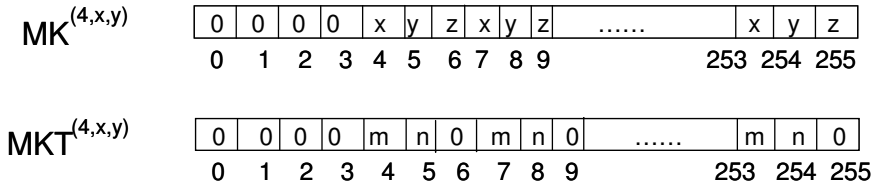

Fig 5: Arrays $MKT^{(4, x, y)}$ along with $MK^{(4, x, y)}$

Where, J belongs to {4, 10, 16, 22….250}. With the proper choice of (x, y, z) the function "t" remains inefficient.

We found many other weak keys of HENKOS. As a last example we want to specify a group of keys which leaves 254 bytes of output keystream at a round to be zero and the remaining 2 bytes with a same nonzero(may also be zero some times) value. Members of such group can be described as below

- All the 128 values at the odd indices of each MK are set to zero.
- Out of the remaining 128 even indices of MK, 127 values are set to zero.

- The value at the remaining even index is L, where L is a member of the set Symmetry.

Now when "t" function acts on an internal state S_i the contents of $S_i[0]$ and $S_i[L]$ are exchanged leaving all the remaining bytes of S_i to be unchanged. Hence, the in the output keystream Z_i (=S_i XOR $t(S_i)$) values at 0^{th} and L^{th} bytes are same and all the other bytes are zero.

4. Conclusion:

In this paper we have shown the weaknesses in the stream cipher HENKOS. The first one is a chosen IV attack which shows a weakness in the IV scheme of the cipher. The second weakness we found is weak keys. We found a set of keys which all lead to same keystream output for a particular value of IV. This shows the weakness in the key initialization and updating internal state and the output function. Hence it is concluded that the stream cipher HENKOS is not secure.

References:

[1] Marius Oliver Gheorghita, HENKOS Stream Cipher: posted to eprint.
(available on www.eprint.iacr.org/2004/080.pdf)