Use collision resistant hash function to build H-MACs

Collisions for the hash function H are distinct inputs x and y such that H(x) = H(y).
A function H is collision resistant if it is infeasible for any probabilistic polynomial-time algorithm to find a collision in H. A family of functions indexed by s is given by

$$H^s(x) = H(s,x)$$

A hash function is a pair of algorithms (Gen, H) where $Gen(1^n)$ outputs the index s (for choosing H^s). If H^s is defined only for inputs x of a certain length, we say it is a fixed length hash function.
A hash function(Gen, H) is collision resistant if for all probabilistic polynomial time adversaries A:

$$Pr[\text{Output of Hash-game} = 1] <= negl(n)$$

H-MACs is a message authentication code where

1. (Gen, h) : A fixed length hash function
2. (Gen, H) : Hash function after applying MD transform to (Gen,h)
3. Fixed constants are IV, opad and ipad

$$\text{HMAC tag for } m = H^s_{IV}((k \oplus opad) \,||\, (H^s_{IV}((k \oplus ipad) \,||\, m))$$

The message is broken into blocks of a predefined size and repeatedly hash them using a fixed length hash function. The input to this hash is the previous block and the current message block.
The initial input is the hash of the two values which are $(k \oplus ipad)$ and IV where k is the key, ipad is the repeated inner pad and IV is the initialization vector.
The final value that is obtained as output is also hashed again with the hash of the two values which are $k \oplus opad$ and IV where opad is the repeated outer pad.

H-MACs are usually derived from NMACs. NMACs require two different keys, whereas HMACs require only a single key which is ‘xor’ed with the inner pad.
The outer pad has only one security requirement which says that the two keys should be different from one another.