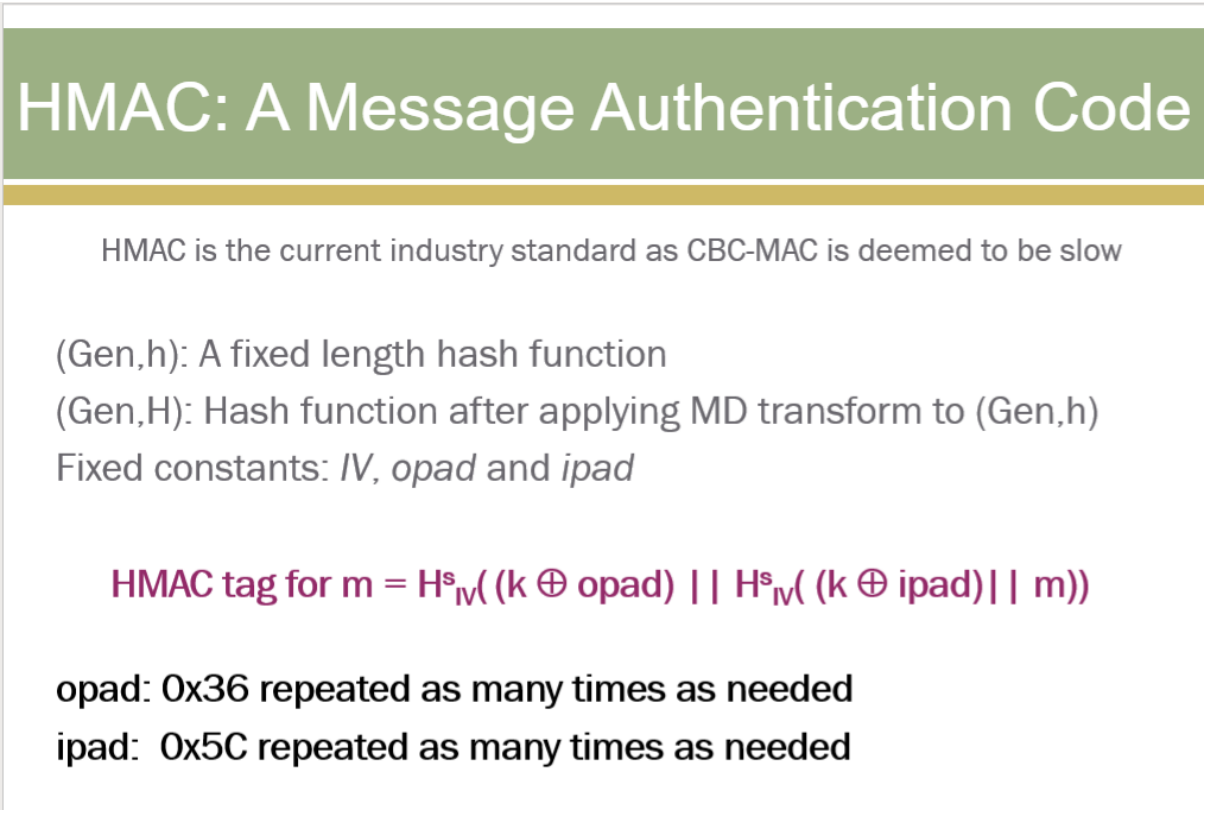
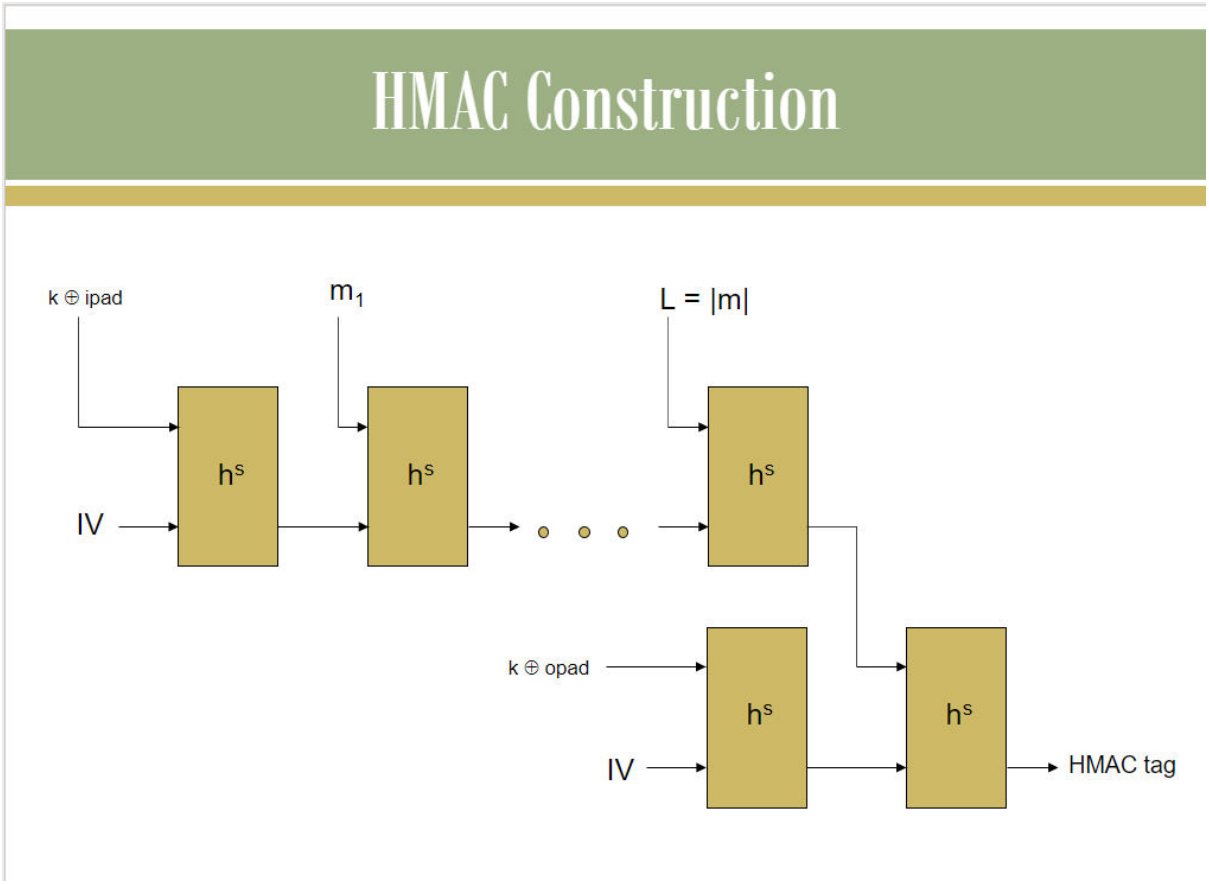
Sudipta Halder (2021202011)

# Use collision resistant hash function to build H-MACs

HMAC is the current industry standard as CBC-MAC is deemed to be slow. We need the help of previously created DLP based fixed length collision resistant hash function. Here it is denoted as $h_s$ in the 2nd picture(HMAC construction) and merkle damgard construction to create HMAC. In HMAC two constants are being used. They are **ipad**=0x36 and **opad**=0x5C.

Now, let's see the construction of HMAC from the below two pictures.






**Now, let's understand how we have implemented this in code.**

We have taken input for prime(p), generator(g), seed to generate h through PRG, key k of length l, initialization vector of length l, where l = no of bits in prime, data of any length.

Working Flow:

1. At first, IPAD and OPAD two constants will be adjusted according to the length of the prime.
2. Then, data is padded with zero if necessary.
3. Then, x1 = k xor IPAD, x2 = initial vector. It is passed through DLP based hash func.
4. Then, a loop is run for d times, where d is no of data blocks.
5. For each iteration, x1 = corresponding msg block, x2 = previous func output.
6. After loop finishes, in the next iteration, x1 = length of data, x2 = previous hash func output, passed into DLP based hash func.
7. In the next iteration, x1 = k xor OPAD, x2 = initial vector. It is passed through DLP based hash func.
8. In the next iteration x1 = output from 6, x2 = output from 7, It is passed through DLP based hash func.
9. The HMAC TAG is the output of 8.