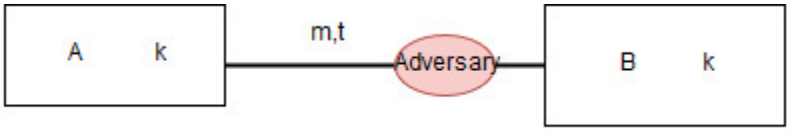
CS558 - February 14, 2017 Lecture Notes

Scribed by: Kevin Leung

MACS



MD5(k||m) = bad MAC, vulnerable to length extension.

Example of a Function, not PERMUTATION

Side Note:

GOOD MAC SCHEME:

$PRF_k(m) = t$

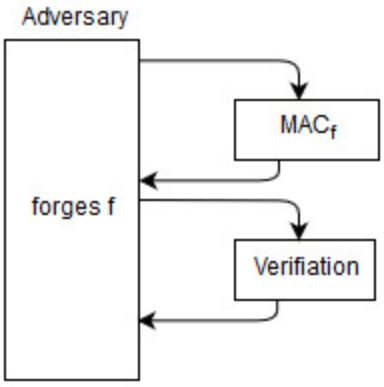
$Ver_k(m,t) = 1$ if PMF = f [success]

0 else [failure]

Given a message and a key, you get the same tag (deterministic) (single input key will always give you the same output).

How could the adversary find a valid t without knowing k_2?

Extension Forgery Against Chosen Message Attacks:

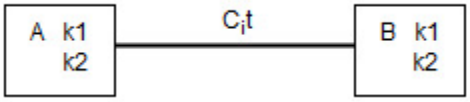


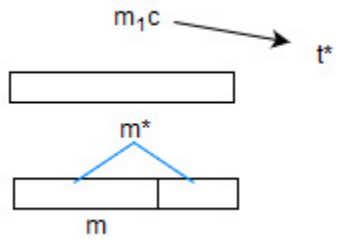
HMAC MD6

~~MD5~~(k XOR string1) || ~~MD5~~((key EXOR string 2) || m))

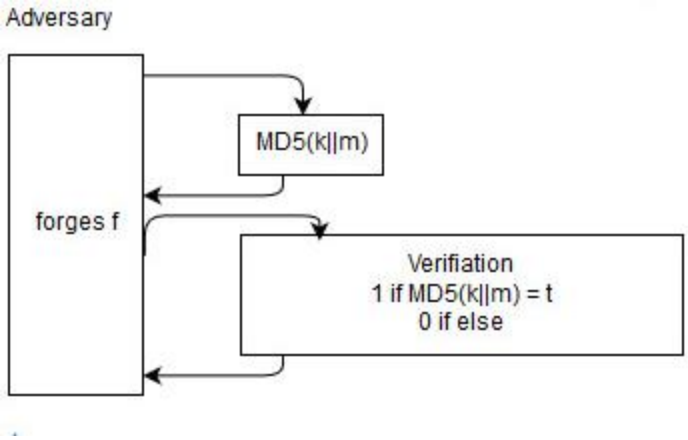
SHA256 SHA256

Usually, instead of MD5, SHA256 is used and it is more secure.





t* can be obtained from m1 and t using length extension technique.



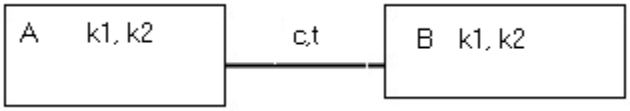
Has access to the Encryption & Decryption Oracle.

If $Ver_{k2}(c,t) = 1$

Output $m' = Dec_{kt}(c)$

Else:

Fail



$c = ENC_k(m)$

$t = MAC_{k2}(c)$

CPA does not have decryption oracle.

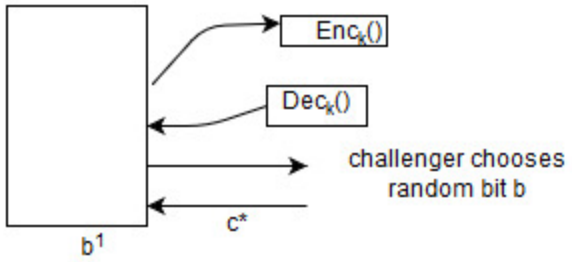
If VERk2(c,t) = 1 (true):

Output m' = DECkt(c)

Else:

Output "Fail"

CCA Security:



Dec is CPA secure.

Adding tag makes the decryption oracle completely useless… AKA it always outputs “fail”.

Achieve CCA Security:

- Let Enc & DeC be CPA security scheme (CBC mode)
- Let MAC be source message authentication code.

Doesn’t output fail -> Adversary has some c,t st $VER_{k2}(c,t) = 1$

Without knowing k, how can we find t?

We can’t from the security of MAC.

MAC prevents CCA attacks.