chap07

[陈永俊] [522031910203]

7.1

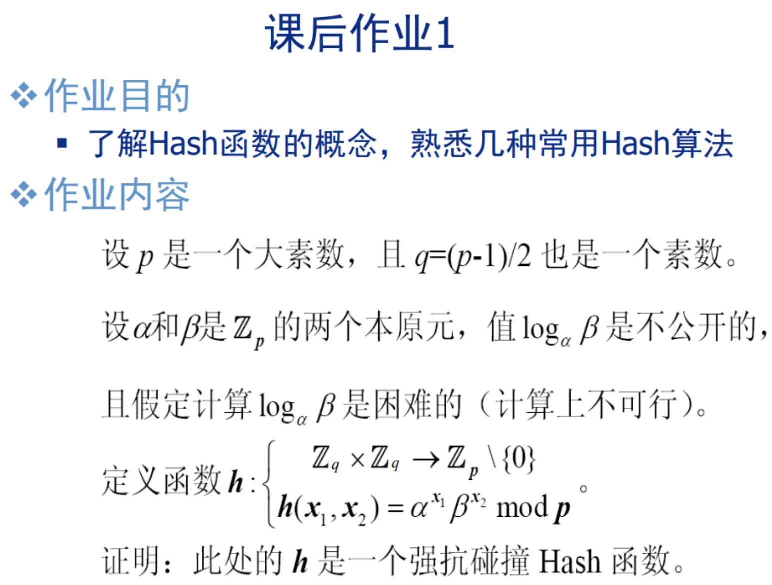

Figure 1: chap07.1

$$
\begin{aligned}
\forall (x_1, x_2): \\
h(y_1, y_2) = h(x_1, x_2) \\
\iff \alpha^{y_1}\beta^{y_2} = \alpha^{x_1}\beta^{x_2}\, mod\, p \\
\iff \alpha^{y_1 - x_1} = \beta^{x_2 - y_2} \\
\iff y_1 - x_1 = (x_2 - y_2)\log_\alpha \beta
\end{aligned}
$$

最后一步需要计算 $\log_\alpha \beta$ 的值，这是不可行的，所以该 **h** 是一个强抗碰撞的 Hash 函数。

7.2

加密算法：

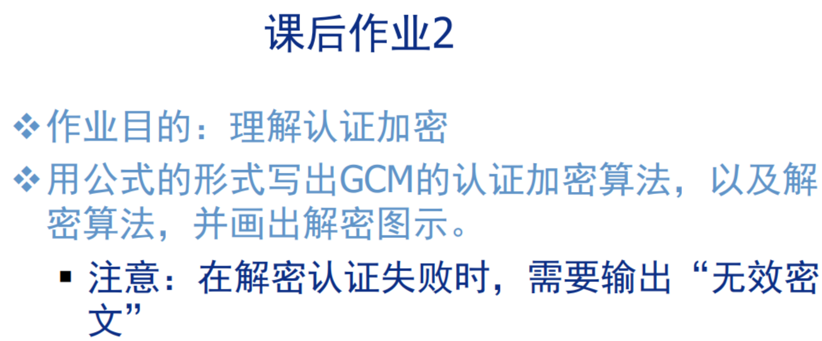

Figure 2: chap07.2

$$X_i = \begin{cases} 0, & \text{for } i = 0 \\ (X_{i-1} \oplus A_i) \cdot H, & \text{for } i = 1, \ldots, m-1 \\ (X_{m-1} \oplus (A_m^* \parallel 0^{128-v})) \cdot H, & \text{for } i = m \\ (X_{i-1} \oplus C_i) \cdot H, & \text{for } i = m+1, \ldots, m+n-1 \\ (X_{m+n-1} \oplus (C_m^* \parallel 0^{128-u})) \cdot H, & \text{for } i = m+n \\ (X_{m+n} \oplus (\mathrm{len}(A) \parallel \mathrm{len}(C))) \cdot H, & \text{for } i = m+n+1. \end{cases}$$

解密算法：

$$\begin{aligned} H &= E(K, 0^{128}) \\ Y_0 &= \begin{cases} IV \parallel 0^{31}1, & \text{if } \mathrm{len}(IV) = 96 \\ \mathrm{GHASH}(H, \{\}, IV), & \text{otherwise.} \end{cases} \\ T' &= \mathrm{MSB}_t(\mathrm{GHASH}(H, A, C) \oplus E(K, Y_0)) \\ Y_i &= \mathrm{incr}(Y_{i-1}) \quad \text{for } i = 1, \ldots, n \\ P_i &= C_i \oplus E(K, Y_i) \quad \text{for } i = 1, \ldots, n \\ P_n^* &= C_n^* \oplus \mathrm{MSB}_u(E(K, Y_n)) \end{aligned}$$

解密流程图：

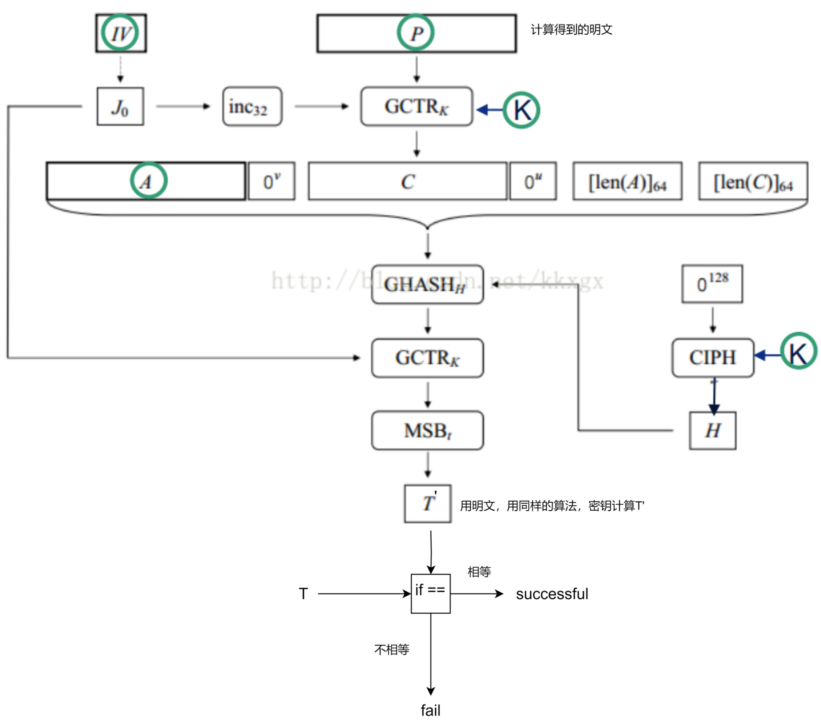
IV
P
计算得到的明文
J0
inc32
GCTRK
K
A
0v
C
0u
[len(A)]64
[len(C)]64
GHASHH
http://blog.csdn.net/kkxgx
0128
GCTRK
CIPH
K
MSBt
H
T'
用明文，用同样的算法，密钥计算T'
T
if ==
相等
successful
不相等
fail

Figure 3: 7.2