Information Security 2020
2nd Project

Prof. Junbeom Hur

TA. Dohyun Ryu

Information System Security Lab.,
Department of Computer Science and Engineering,
Korea University, Seoul, Korea

KOREA UNIVERSITY ISSLAB Information System Security Laboratory

Hash Function

- Consider the following structure of a hash function that takes a 768-bit input and returns a 256-bit output

input : X = $(A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0, W_0, W_1, \ldots\ldots\ldots\ldots W_{15})$.
Where $A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0, W_0, W_1, \ldots\ldots\ldots\ldots W_{15}$ are each 32-bit.

Intermediate variable : $A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i, W_i$ each are 32-bit.

number of Round: R

#output : Y = $(A_0 \oplus A_i, B_0 \oplus B_i, C_0 \oplus C_i, D_0 \oplus D_i, E_0 \oplus E_i, F_0 \oplus F_i, G_0 \oplus G_i, H_0 \oplus H_i)$

For (i = 16; i<R; i++):

$$W_i = (W_{i-3} <<< 1) \oplus (W_{i-8} <<< 6) \oplus (W_{i-14} <<< 11) \oplus W_{i-16}$$

For (i = 0; i<R; i++):

$(A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1}, E_{i+1}, F_{i+1}, G_{i+1}, H_{i+1})$ = Round$(A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i, W_i)$

Hash Function

- i-th *Round()* function structure

input : $(A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i, W_i)$

Intermediate variable : T which is 32-bit

output : $(A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1}, E_{i+1}, F_{i+1}, G_{i+1}, H_{i+1})$

T = $(W_i[0], W_i[1], W_i[2], W_i[3])$
T = MDS(S(T[0] $\oplus$ $A_i[0]$), S(T[1] $\oplus$ $A_i[1]$), S(T[2] $\oplus$ $A_i[2]$), S(T[3] $\oplus$ $A_i[3]$))
T = MDS(S(T[0] $\oplus$ $B_i[0]$), S(T[1] $\oplus$ $B_i[1]$), S(T[2] $\oplus$ $B_i[2]$), S(T[3] $\oplus$ $B_i[3]$))
T = MDS(S(T[0] $\oplus$ $C_i[0]$), S(T[1] $\oplus$ $C_i[1]$), S(T[2] $\oplus$ $C_i[2]$), S(T[3] $\oplus$ $C_i[3]$))
T = MDS(S(T[0] $\oplus$ $D_i[0]$), S(T[1] $\oplus$ $D_i[1]$), S(T[2] $\oplus$ $D_i[2]$), S(T[3] $\oplus$ $D_i[3]$))
T = MDS(S(T[0] $\oplus$ $E_i[0]$), S(T[1] $\oplus$ $E_i[1]$), S(T[2] $\oplus$ $E_i[2]$), S(T[3] $\oplus$ $E_i[3]$))
T = MDS(S(T[0] $\oplus$ $F_i[0]$), S(T[1] $\oplus$ $F_i[1]$), S(T[2] $\oplus$ $F_i[2]$), S(T[3] $\oplus$ $F_i[3]$))
T = MDS(S(T[0] $\oplus$ $G_i[0]$), S(T[1] $\oplus$ $G_i[1]$), S(T[2] $\oplus$ $G_i[2]$), S(T[3] $\oplus$ $G_i[3]$))

$(A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1}, E_{i+1}, F_{i+1}, G_{i+1}, H_{i+1}) = (H_i \oplus \mathrm{T}, A_i, B_i, C_i, D_i, E_i, F_i, G_i)$

Definitions

- $\oplus$: Bitwise exclusive OR
- <<< n: n-bit left rotation
- S(): S-box of AES (reference: FIPS 197)
- MDS(): Matrix multiplication of AES (reference: FIPS 197)
- T = (T[0],T[1],T[2],T[3]): Concatenates four 8-bit T[0], T[1], T[2], T[3], and assigns the 32-bit result to T
- (T[0],T[1],T[2],T[3]) = T: Separates 32-bit T into four 8-bit T[0], T[1], T[2], T[3]

Goals

1. Given an arbitrary 256-bit output Y, find the maximum number of rounds (R in the previous algorithm) to determine 768-bit input X with less than 2^{253} hash operations

2. Describe the algorithm (for each round) to solve it

Grading

1. Source code and exe file for solution **(20 points)**
   - You don't need to implement an algorithm that requires more than 2^{32} hash operations
   - However, it must be described in the report!
   - Your program has to print X for each round given a value Y
2. Report **(30 points)**
   - You need to show your solutions step by step
   - Appendix: Source code with comments

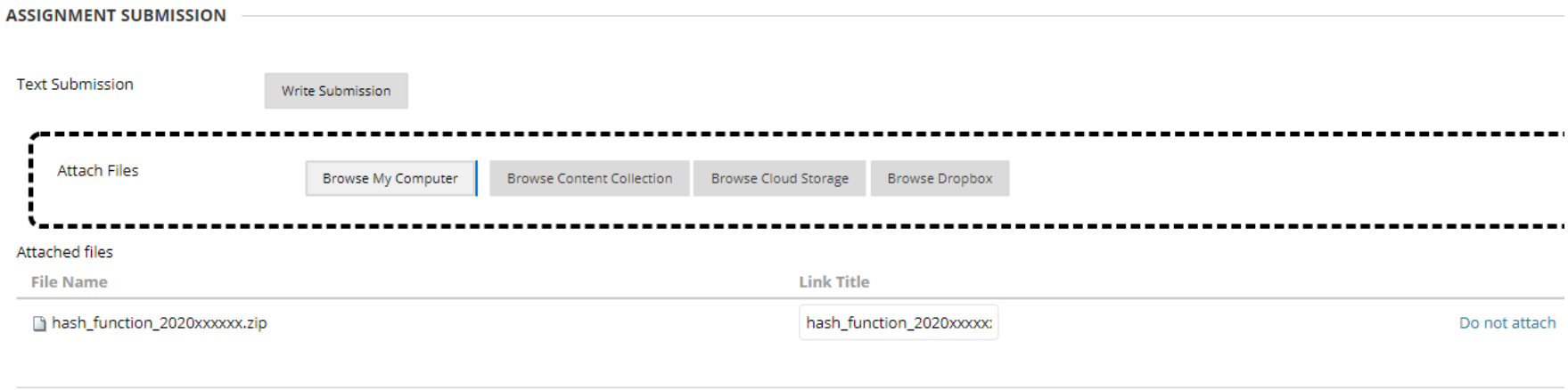
Submission Guideline

- **Please upload the followings on Blackboard**

1. Source code and exe file for solution **(C is encourage, but if you want you can use Python, Java…etc)**
2. Report **(.doc, .hwp, or pdf file)**

→ Compress all of the files **(.zip)**

  - **Late submission and any kind of plagiarism will result in 0 point**



KOREA UNIVERSITY ISSLAB Information System Security Laboratory

Submission Guideline

- **Deadline: 2020 Oct. 31, 23:59:00**
- **Late submission is not accepted**