|  |  |  |  |
| --- | --- | --- | --- |
| **Ex. No. 07** | **HMAC Using SHA-512** | | |
| Date of Exercise | 03-03-2015 | Date of Output Verification | 03-03-2015 |

**Question**

Design a HMAC using SHA-512 hashed function in java

**Procedure**

1. Append zeros to the end of K to create a B byte string (e.g., if K is of length 20 bytes and B=64, then K will be appended with 44 zero bytes 0x00)

2. XOR (bitwise exclusive-OR) the B byte string computed in step (1) with ipad

3 Append the stream of data 'text' to the B byte string resulting from step (2)

4. Apply H to the stream generated in step (3)

5. XOR (bitwise exclusive-OR) the B byte string computed in step (1) with opad

6. Append the H result from step (4) to the B byte string resulting from step (5)

7. Apply H to the stream generated in step (6) and output the result

**Program**

import java.math.BigInteger;

import java.security.MessageDigest;

import java.util.Scanner;

//@author William Scott

public class NetworkSecurityLabExp7HMAC {

//Network Security Lab - Exp 7 - HMAC Using SHA-512

public static String p, k, s, messageip, haship, ipad = "00110110", opad = "01011100";

public static Scanner in = new Scanner(System.in);

public static void main(String[] args) throws Exception {

System.out.println("UR12CS135 - HMAC Using SHA-512");

getvalues();

p = calculate(ipad, messageip);

System.out.println("First Hash Output : " + p);

p = calculate(opad, p);

System.out.println("Output of HMAC (SHA-512) : " + p);

}

public static void getvalues() {

System.out.print("\nEnter 8 bit Key(k) : ");

k = in.nextLine();

System.out.print("Enter Input Message : ");

messageip = in.nextLine();

}

public static String calculate(String pad, String message) throws Exception {

s = xor(k, pad);

haship = Integer.toHexString(Integer.parseInt(s, 2)) + message;

System.out.println("\nAppended Input to SHA-512 : " + haship);

return sha512(haship);

}

public static String xor(String a, String b) {

String c = "";

for (int i = 0; i < a.length(); i++) {

if (a.charAt(i) == '0' && b.charAt(i) == '0') {

c += "0";

} else if (a.charAt(i) == '0' && b.charAt(i) == '1') {

c += "1";

} else if (a.charAt(i) == '1' && b.charAt(i) == '0') {

c += "1";

} else if (a.charAt(i) == '1' && b.charAt(i) == '1') {

c += "0";

}

}

return c;

}

public static String sha512(String a) throws Exception {

MessageDigest sha = MessageDigest.getInstance("SHA-512");

sha.update(a.getBytes());

return new BigInteger(1, sha.digest()).toString(16);

}

}

**Input**

ipad : 00110110 ; opad : 01011100

Enter 8 bit Key (k) : 11110000

Enter Input Message : hello

**Output**

UR12CS135 - HMAC Using SHA-512

Enter 8 bit Key(k) : 11110000

Enter Input Message : hello

Appended Input to SHA-512 : c6hello

First Hash Output : 1d05c925230db41d4fc846e6e9c698305712cb84982384f0f2dcc1b10f0df4b2e3586dc6c6974cc02bdabd8f44f8c5ce60062d1672c67b162597a60d450ff0cb

Appended Input to SHA-512 : ac1d05c925230db41d4fc846e6e9c698305712cb84982384f0f2dcc1b10f0df4b2e3586dc6c6974cc02bdabd8f44f8c5ce60062d1672c67b162597a60d450ff0cb

Output of HMAC (SHA-512) : f3ea759d45a896e334e886f8ea6fca79b56d88f790bb4823831f6d4c9c2f702f65e5b0e98c1ede62348533217f18dc93f6cea70b157c0fa002de0eab76f6e688

**Result**

The implementation of HMAC using SHA512 hashed function is successfully done.

[Signature of the Staff In-charge]

Name of the Staff In – charge: Mr. Manoj Kumar

Date: 03-03-2015