ACM Winter Session for Cyber Security
Practical Problems-Part I

1. Write a program that can encrypt and decrypt using the general Caesar cipher, also known as an additive cipher

2. Create software that can encrypt and decrypt using a 2 * 2 Hill cipher

3. Create software that can encrypt and decrypt using a general substitution block cipher.

4. Create software that can encrypt and decrypt using S-DES. Test data: use plaintext, ciphertext, and decrypt the string 01000110 using the key 1010000010 by hand. Show intermediate results after each function (IP, FK, SW, FK, IP-1). Then decode the first 4 bits of the plaintext string to a letter and the second 4 bits to another letter where we encode A through P in base 2 (i.e., A = 0000, B = 0001, c, P = 1111).
Hint: As a midway check, after the xoring with K2, the string should be 11000001

5. Create software that can encrypt and decrypt using S-AES. *Test data*: A binary plaintext of 0110 1111 0110 1011 encrypted with a binary key of 1010 0111 0011 1011 should give a binary ciphertext of 0000 0111 0011 1000. Decryption should work correspondingly.

6. Create software that can encrypt and decrypt in cipher block chaining mode using one of the following ciphers: affine modulo 256, Hill modulo 256, S-DES, DES.
Test data for S-DES using a binary initialization vector of 1010 1010. A binary plaintext of 0000 0001 0010 0011 encrypted with a binary key of 01111 11101 should give a binary plaintext of 1111 0100 0000 1011. Decryption should work correspondingly.

7. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
8. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

9. Write a Java program to implement RSA Algorithm.