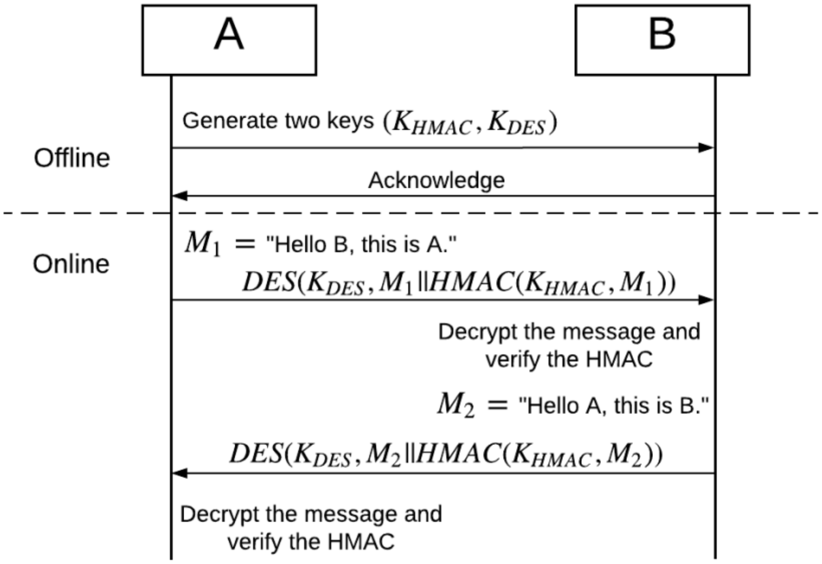
# Lab 2: Implementation and Application of HMAC

HMAC is a keyed-hash type of message authentication code (MAC), involving a hash function and a secret key. It can simultaneously provide the data integrity and the authentication of a message. According to the different underlying hash functions MD5, SHA-1, SHA-256, etc., the algorithm is termed HMAC-MD5, HMAC-SHA1, HMAC-SHA256, etc.

## Task

Two students in a group. Student A and B share a key for HMAC in an offline manner. Student A then generates a message, gets the HMAC digest of this message, and then sends the message along with its HMAC to student B. Student B verifies the integrity of the received message by generating another HMAC with the shared key and match the two HMACs. Students A and B switch the roles and do the above again. All the transmitted messages should be encrypted with DES.

A simple protocol for the above process may be like


A
B
Generate two keys $(K_{HMAC}, K_{DES})$
Offline
Acknowledge
Online
$M_1$ = "Hello B, this is A."
$DES(K_{DES}, M_1 \| HMAC(K_{HMAC}, M_1))$
Decrypt the message and verify the HMAC
$M_2$ = "Hello A, this is B."
$DES(K_{DES}, M_2 \| HMAC(K_{HMAC}, M_2))$
Decrypt the message and verify the HMAC

**Both sides should display:**

1. Shared keys for HMAC and DES
2. Plain message to be sent
3. HMAC
4. Ciphertext to be sent
5. Received ciphertext
6. Plain message and HMAC after decryption
7. Generated HMAC using $K_{HMAC}$ and plain message

## Submission

Demo your program to the TA (70);

Submit a lab report to Canvas including two parts: 1. what you have learned about HMAC (10); 2. steps of your work (e.g., the language and library you used, problems you encountered and how you solved them, etc.) Screen captures is recommended (20). The report should be an individual work, and don't forget to mention who is your partner.

## Due

Demo: 5pm on Feb. 21

Report: 5pm on Feb. 28