Announcement

- Project 1 is due at 7PM on Tuesday, 3/8.
- The API for the library you should implement in Project 1

```
public class Crypto
-------------------------------------------------------------
    int[] DES(int[] plaintext, int[] key)
    int[] ECB(String plaintext, String key)
    int[] CBC(String plaintext, String key, String IV)
```

- Homework 2 is due at 7PM on Tuesday, 3/15.

Last Time

- Message Authentication Code: $MAC(k, m) = h(m|k)$
- Public key cryptography
  - A pair of keys: a public key k^+ and a private key k^-
  - $D(E(m, k^+), k^-) = m; D(E(m, k^-), k^+) = m$
- RSA

Encryption using private key

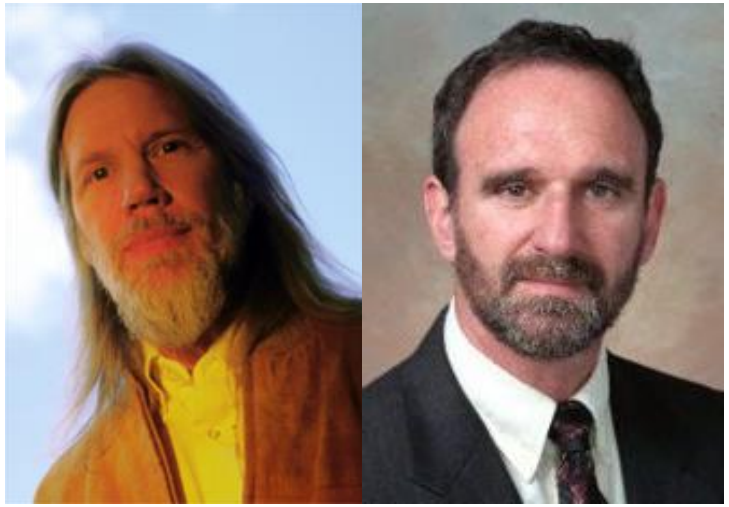
- What if we use private key k^- for encryption and public key k^+ for decryption
  - Encryption algorithm: $c = E(k^-, m) = m^d \bmod n$
  - Decryption algorithm: $m = D(k^+, c) = c^e \bmod n$
- Example:
  - Public key $k^+ = <7, 33>$, Private key $k^- = <3, 33>$
  - Plaintext $m = 4$
  - Encryption: $c = E(k^-, m) = 4^3 \bmod 33 = 64 \bmod 33 = 31$
  - Decryption: $m = D(k^+, c) = 31^7 \bmod 33 = 27512614111 \bmod 33 = 4$
- Q: Why encrypt with private key?

Digital Signature

- Model physical signatures in digital world
  - Create association between private key and document
  - Provide data integrity and non-repudiation
- Sign a document
  - Given document *m*, private key k^-
  - Signature $S(m) = E(k^-, h(m))$
- Validation
  - Given document *m*, signature *S(m)*, public key k^+
  - Validate $D(k+, S(m)) = h(m)$

Diffie-Hellman

- How can two participants negotiate a secret key if there is no secure channel?
- Diffie-Hellman
  - The first public key cryptosystem, but dose neither encryption nor digital signature
  - Used to negotiate a shared secret key over an insecure media
- Mathematics behind: it is computationally hard to calculate discrete logarithm
  - Given g and g^x, it is computationally hard to find x



Diffie-Hellman Protocol

- Diffie-Hellman Key Agreement
  - Setup: two participants A and B agree on a prime number p and a base $g(<p)$ (p and g are public)
  - STEP1: A picks a private value $SA(<p-1)$, B picks a private value $SB(<p-1)$
  - STEP2: A generates $T_A = g^{SA} \bmod p$, B generates $T_B = g^{SB} \bmod p$. A and B exchange T_A and T_B.
  - STEP3: A computes $T_B^{SA} \bmod p$, B computes $T_A^{SB} \bmod p$. The shared secret key $k = T_B^{SA} \bmod p = T_A^{SB} \bmod p$ (why?)

    $T_B^{SA} = (g^{SB})^{SA} = g^{SB \cdot SA} = g^{SA \cdot SB} = (g^{SA})^{SB} = T_A^{SB} \bmod p$

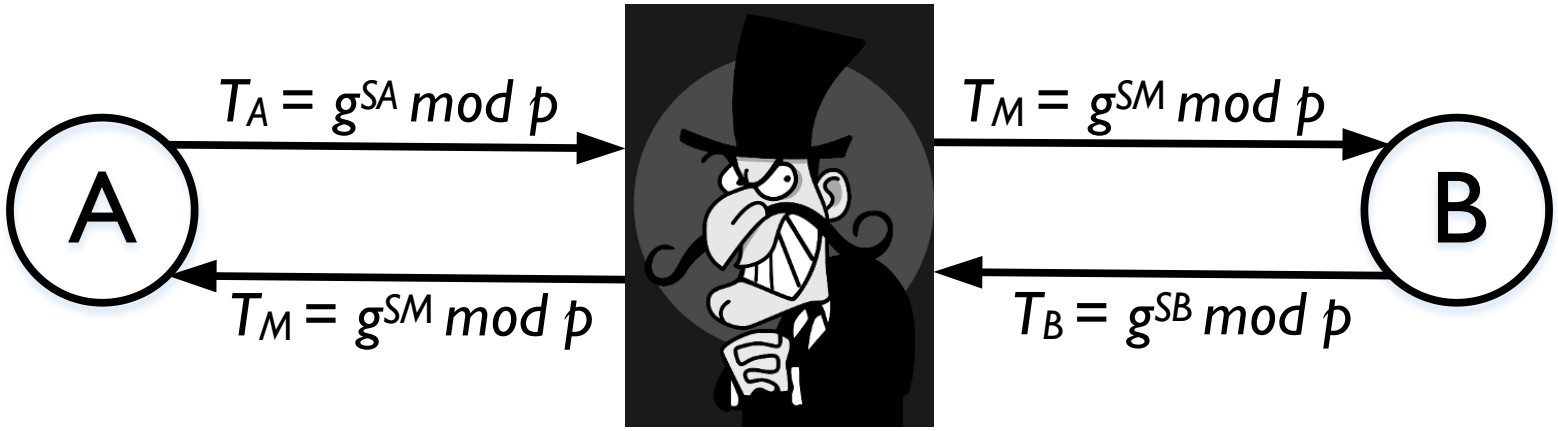
Exercise

- Diffie Hellman + Caesar Cipher
  - SETUP: Two students form a group. One is sender, the other is receiver
  - STEP1: Sender and receiver use Diffie-Hellman to agree on a secret key *k*
  - STEP2: If *k*>25, use *k mod* 25 as the secret key
  - STEP3: The sender generates a message and encrypts it using Caesar cipher with secret key *k*. Then, the sender passes the ciphertext to the receiver.
  - STEP4: The receiver decrypts the ciphertext with the secret key *k*

Caesar Cipher: Index the letters in alphabet from 0 to 25. Each ciphertext letter c = E(m, k) = (m+k) mod 26.

Man-in-The-Middle Attack

- You really don't know anything about who you have exchanged keys with
- Man-in-the-middle



  - A and B think they are talking directly to each other, but the attacker is actually performing two separate exchanges
- Solution: authentication

Meet Alice and Bob

- Alice and Bob are the canonical players in the cryptographic world
  - They represent the end points of some interaction
  - Used to illustrate/define a security protocol
- Other players occasionally join
  - Mallory: malicious entity
  - Eve: eavesdropper
  - Trent: trusted third party

Notation

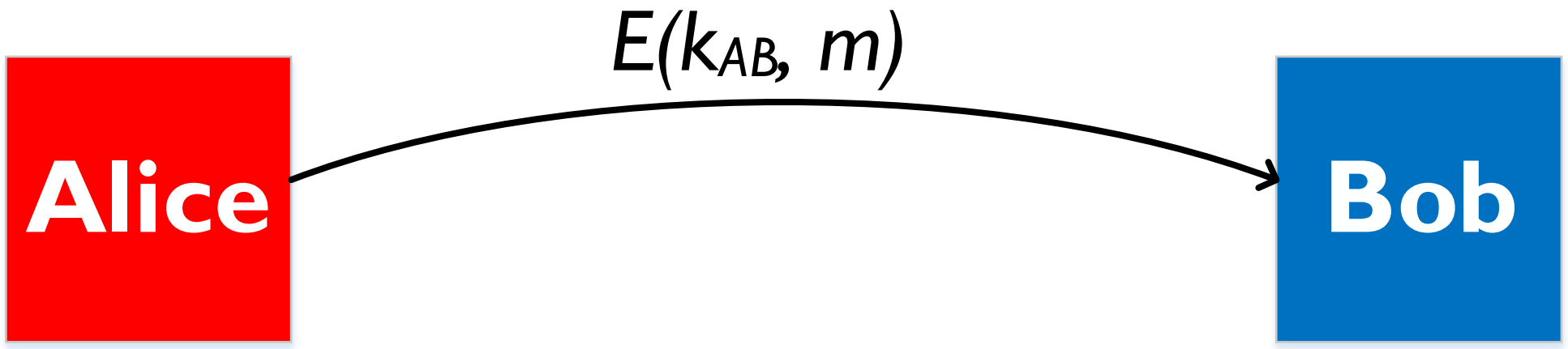
- You will generally see protocols defined in terms of exchanges containing some notation like
  - All players are identified by their first initial, e.g., Alice = *A*, Bob = *B*
  - Data or message is *m*
  - k_{AB} is a symmetric key known to *A* and *B*
  - k_A^+ and k_A^- is a public/private key pair for *A*
  - *E(k, m)* is encryption of data *m* with key *k*
  - *h(m)* is the hash of data *m*
  - $Sig(k_A^-, m)$ is the digital signature (using *A*'s private key) of data *m*
  - pw^A is the password for *A*

Security in Communication

- when communicating
  - Ensure the authenticity of a user
  - Ensure the integrity of the data
  - Keep data confidential
  - Guarantee non-repudiation

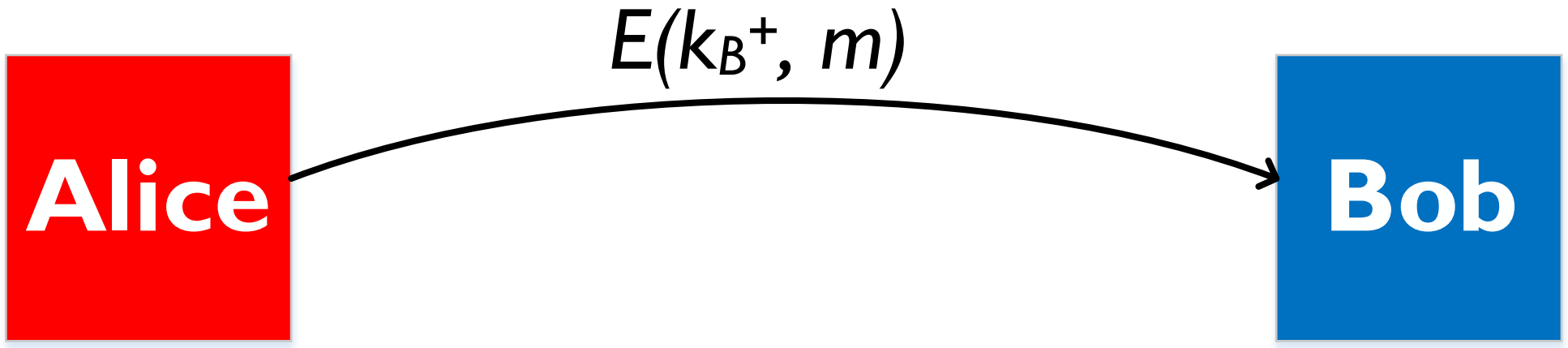
Data Confidentiality

- Alice wants to ensure that the data is not exposed to anyone except the intended recipient Bob
  - Secret key cryptography, i.e., DES, CBC



Data Confidentiality

- Alice wants to ensure that the data is not exposed to anyone except the intended recipient Bob
  - Public key cryptography, i.e., RSA

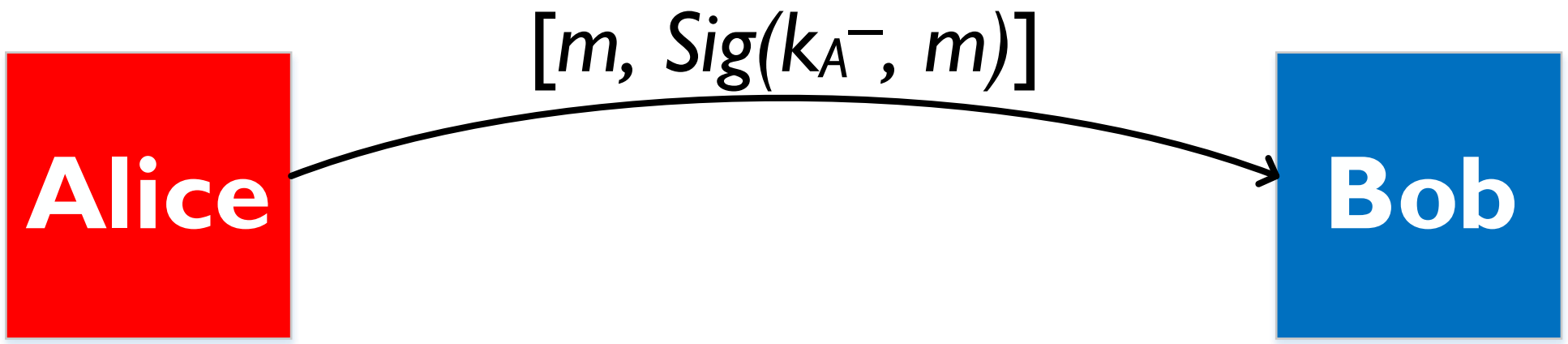


Data Integrity

- Alice wants to ensure that any modification of the data in flight is detectable by Bob
  - Secret key cryptography, i.e., message authentication code

Data Integrity

- Alice wants to ensure that any modification of the data in flight is detectable by Bob
  - Public key cryptography, i.e., digital signature



Non-repudiation

- Non-repudiation: the ability to confirm that a message is generated by a particular person
- Data Integrity vs. Non-repudiation
  - If the integrity of data is preserved by using message authentication code, is it provably from that source?
  - What about if we use digital signature instead?