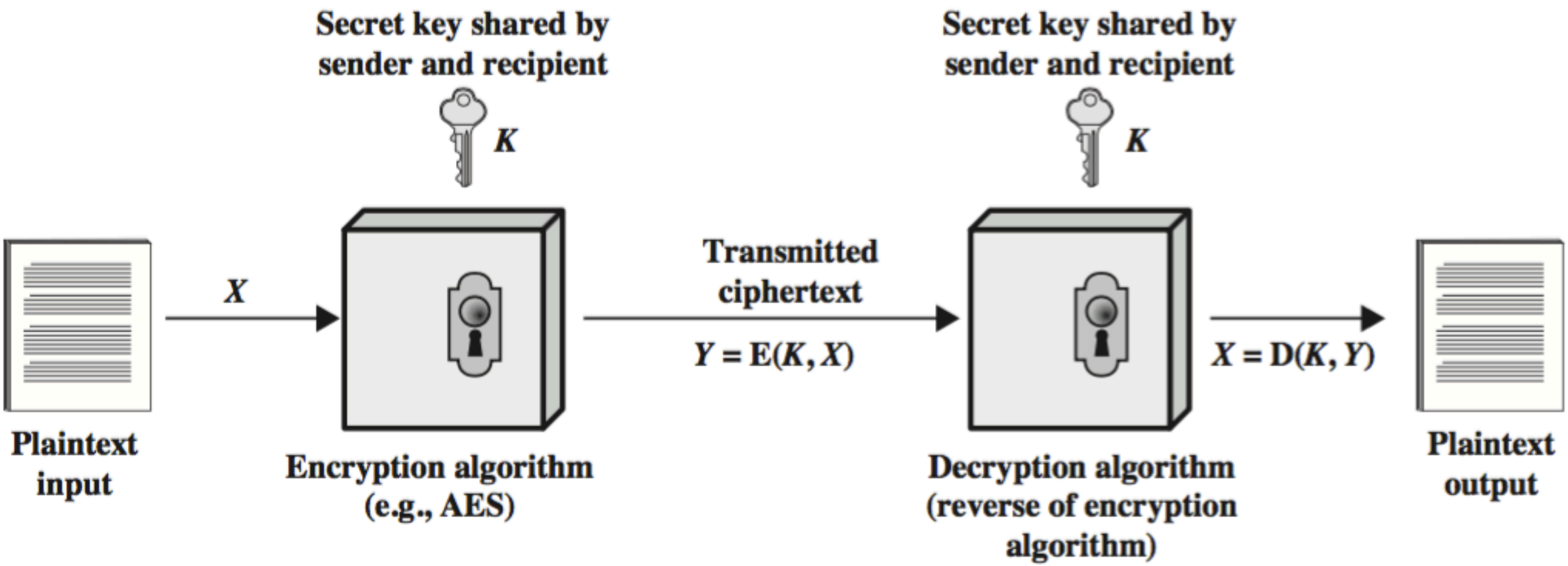
# Crash course in Cryptography (for 1st lab)

- Three basic ingredients
  - **Symmetric encryption**
  - **Public key cryptography**
  - **Message digests** (cryptographic hash functions)

# Symmetric Encryption

# Symmetric Encryption



# Symmetric Encryption

- Also known as Conventional Encryption
- Sender and receiver use <u>same</u> key (shared secret)
- Was the only method used prior to the 1970s & still the main “workhorse”
- Popular algorithms:
  - Advanced Encryption Standard (AES)
  - Triple Data Encryption Standard (3DES)
  - Rivest Cipher 4 (RC4)
- Fast
- But how to share secret keys?
  - “chicken-and-egg” problem

# Public Key Encryption

# Public Key Cryptography

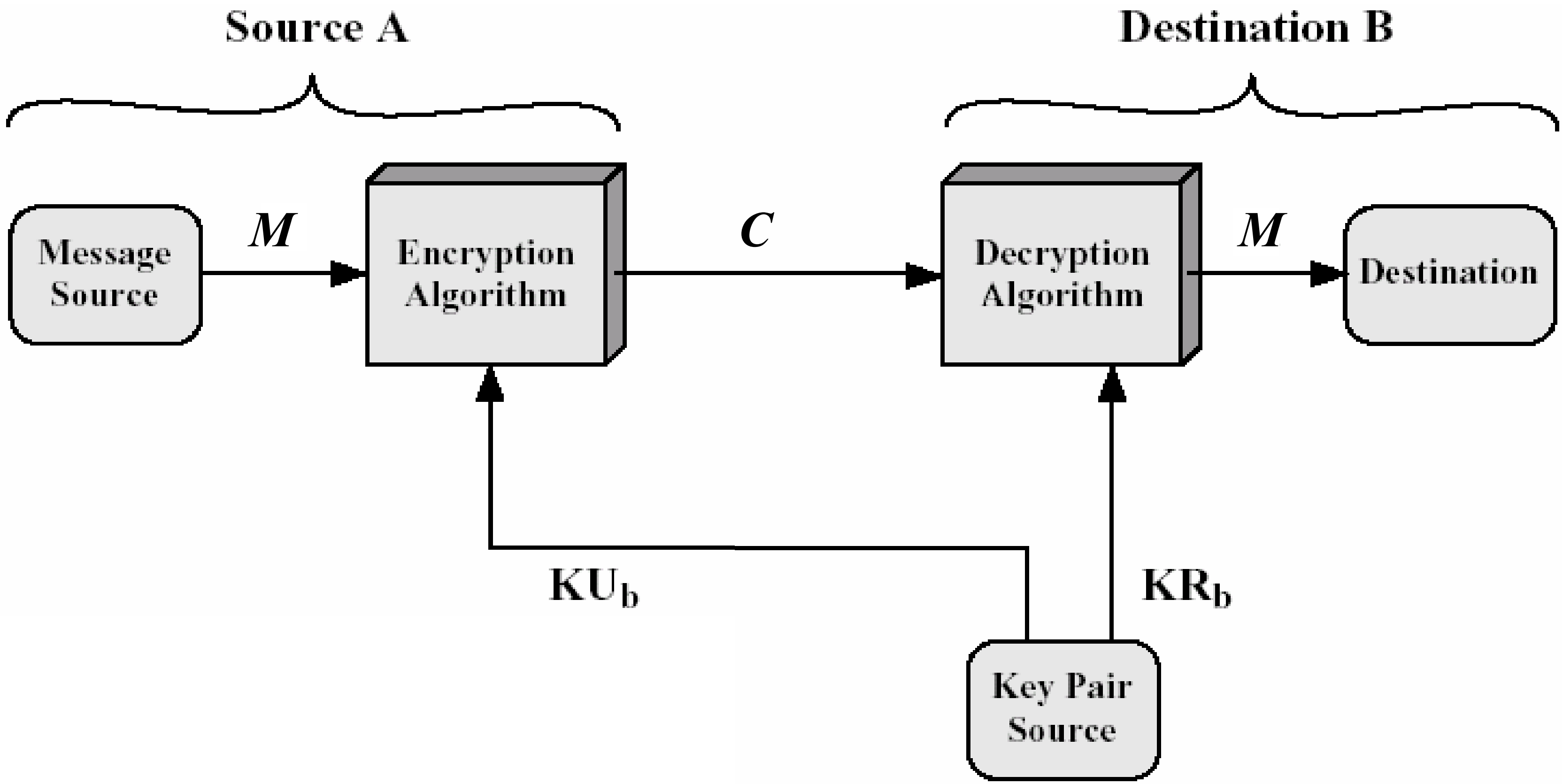
- Each party has two keys:
  - a **public key**, known potentially to anybody, used to **encrypt messages**, and **verify signatures**
  - a **private key**, known only to its owner, used to **decrypt messages**, and **create signatures**

- Complements rather than replaces symmetric cryptography
  - Used for exchanging secret keys

# Application: Secrecy

- Alice (A) sends message to Bob (B) by encrypting with his public key
- Message can only be decrypted with Bob's corresponding private key (known only to him)

# Secrecy Model




Encryption: $C = E_{KUb}(M)$

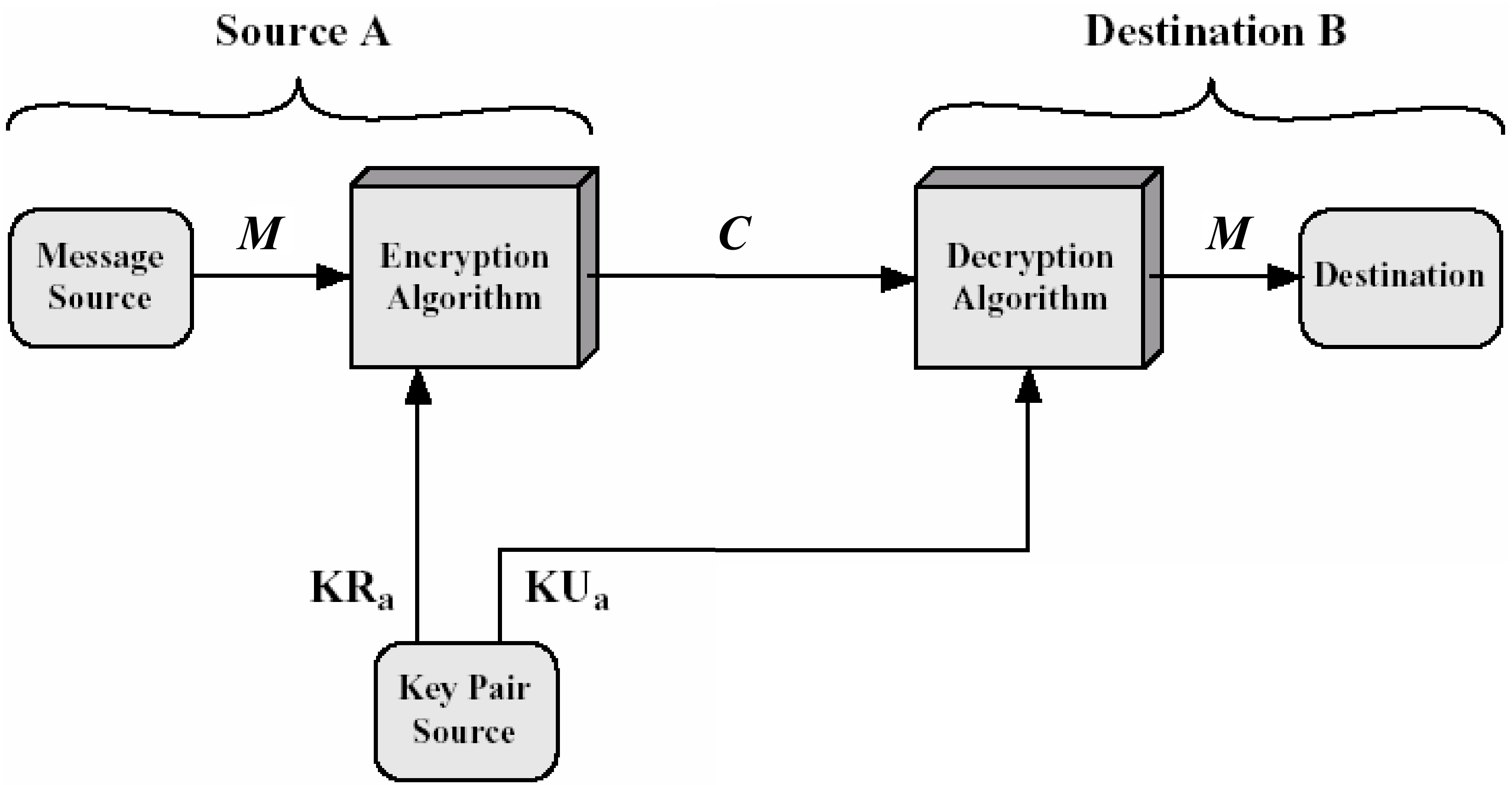
Decryption: $M = D_{KRb}(C) = D_{KRb}(E_{KUb}(M))$

# Application: Authentication

- Alice (A) sends message to Bob (B) encrypting it with her own private key (i.e. she signs the message)

- Everyone with Alice's public key can decrypt the message. A message that can be decrypted with Alice's public key ***must have come from Alice***.

# Authentication Model




Signing: $C = E_{KRa}(M)$

Verifying: $M = D_{KUa}(C) = D_{KUa}(E_{KRa}(M))$

# Authenticity of Public Keys: MITM attack

- Bob’s public key is in the public domain and only Bob has the corresponding private key
- What happens though if an eavesdropper (Eve) generates another key pair and advertises the public key produced as belonging to Bob?
- People then may send messages to Bob using the wrong public key, for which Eve has the corresponding private key.

⇒ *Need to be able to* ***trust*** *that a public key belongs to whom it’s reputed to belong.*

# Cryptographic Hash Functions

# Data Integrity

- Assurance of non-alteration
- CRC or checksums (as implemented in TCP, say) are designed to detect accidental bit errors due to noise, etc.
  - Not enough to withstand a deliberate attack

# Cryptographic Hash Function

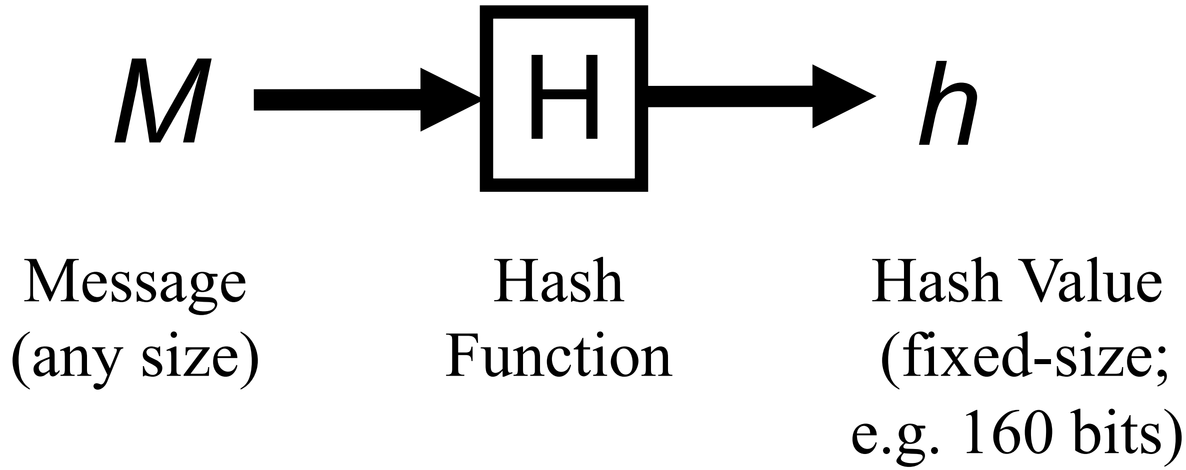
- Used to provide integrity of a message
- Purpose is to produce a fixed-size *hash-value:*

$$h = H(M)$$

where $h$ is the hash value
$H$ is the hash function
$M$ is the message

- Any change in $M$, however small, should produce a different $h$-value

# Cryptographic Hash Function




- Note that a hash function is a many-to-one function. Potentially many messages can have the same hash, but finding these should be very difficult