CS420: Operating Systems

Cryptography

James Moscola
Department of Engineering & Computer Science
York College of Pennsylvania

Based on Operating System Concepts, 9th Edition by Silberschatz, Galvin, Gagne

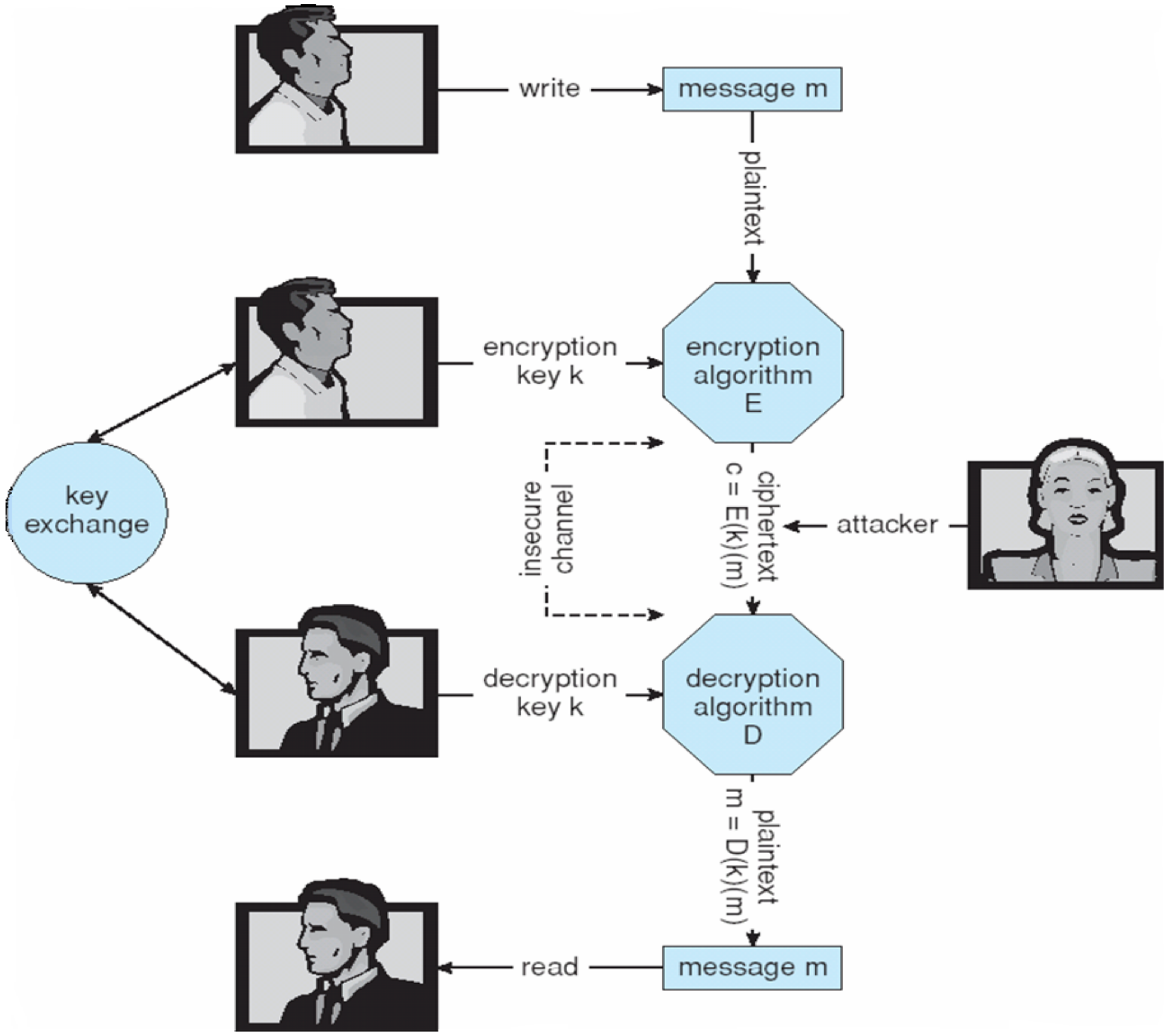
Cryptography as a Security Tool

- **Broadest security tool available when communicating with unknown users over a network**
  - An eavesdropper may be listening to your communication
    - **Encryption** can prevent an eavesdropper from reading messages
  - Difficult to know what machine you are communicating with over network
    - **Authentication** can confirm the identity of remote machine

Cryptography as a Security Tool (cont.)

- **Means to constrain potential senders (sources) and / or receivers (destinations) of messages**
  - Based on secrets (keys)
  - Enables
    - Confirmation of source
    - Receipt of message only by certain destination
    - Trust relationship between sender and receiver

Secure Communication Over Insecure Medium



Need a secure way to exchange keys

Encryption

- **Encryption algorithm consists of**
  - Set K of keys
  - Set M of Messages
  - Set C of ciphertexts (encrypted messages)
  - A function $E : K \to (M \to C)$. That is, for each $k \in K$, $E(k)$ is a function for generating ciphertexts from messages
    - Both E and E(k) for any k should be efficiently computable functions
  - A function $D : K \to (C \to M)$. That is, for each $k \in K$, $D(k)$ is a function for generating messages from ciphertexts
    - Both D and D(k) for any k should be efficiently computable functions
- **An encryption algorithm must provide this essential property: Given a ciphertext $c \in C$, a computer can compute m such that $E(k)(m) = c$ only if it possesses D(k)**
  - Thus, a computer holding D(k) can decrypt ciphertexts to the plaintexts used to produce them, but a computer not holding D(k) cannot decrypt ciphertexts
  - Since ciphertexts are generally exposed (for example, sent on the network), it is important that it be infeasible to derive D(k) from the ciphertexts

Symmetric Encryption

- **Same key used to encrypt and decrypt**
  - E(k) can be derived from D(k), and vice versa
  - Requires sender and receiver to know a shared secret key
- **DES is a commonly used symmetric block-encryption algorithm**
  - Encrypts a block of data at a time
  - Triple-DES considered more secure (DES applied 3 times)
- **Advanced Encryption Standard (AES) is widely used (favored by many)**

Diffie Hellman Key Exchange

- **Does not require sender / receiver know a shared secret key**
  - Sender and receiver each have two keys: a shared public key and a private key
  - These keys are used to create a shared secret key
- **Does NOT provide authentication**
- **Vulnerable to man-in-the-middle attacks**
  - Another public key cryptography technique that avoids this problem is RSA

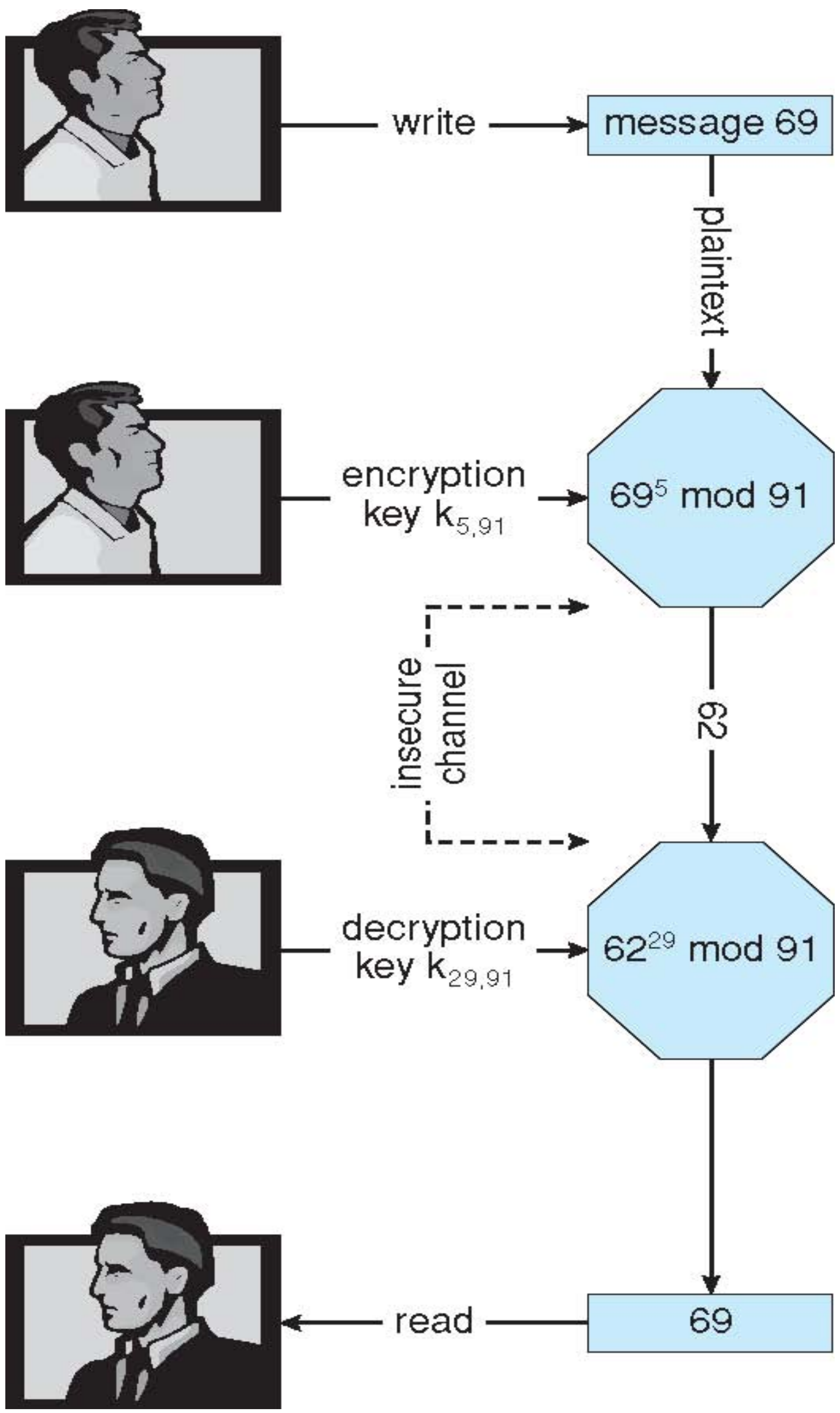
Asymmetric Encryption

- **Public-key encryption based on each user having two keys:**
  - **shared public key** – published key used to encrypt data
  - **private key** – key known only to individual user used to decrypt data
- **Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme**
  - RSA block cipher is common
  - Efficient algorithm for testing whether or not a number is prime
  - No efficient algorithm is know for finding the prime factors of a number

Asymmetric Encryption (Cont.)

- **Formally, it is computationally infeasible to derive $D(k_d , N)$ from $E(k_e , N)$, and so $E(k_e , N)$ need not be kept secret and can be widely disseminated**
  - $E(k_e , N)$ (or just k_e) is the public key
  - $D(k_d , N)$ (or just k_d) is the private key
  - N is the product of two large, randomly chosen prime numbers p and q (for example, p and q are 512 bits each)
  - Encryption algorithm is $E(k_e , N)(m) = m^{k_e} \bmod N$, where k_e satisfies $k_e k_d \bmod (p-1)(q-1) = 1$
  - The decryption algorithm is then $D(k_d , N)(c) = c^{k_d} \bmod N$

Encryption/Decryption with RSA Asymmetric Cryptography



Authentication

- **Constraining set of potential senders of a message**
  - Complementary and sometimes redundant to encryption
  - Also can prove message unmodified
- **Algorithm components**
  - A set K of keys
  - A set M of messages
  - A set A of authenticators
  - A function $S : K \rightarrow (M \rightarrow A)$
    - That is, for each $k \in K$, $S(k)$ is a function for generating authenticators from messages
    - Both S and $S(k)$ for any k should be efficiently computable functions
  - A function $V : K \rightarrow (M \times A \rightarrow \{\text{true, false}\})$. That is, for each $k \in K$, $V(k)$ is a function for verifying authenticators on messages
    - Both V and $V(k)$ for any k should be efficiently computable functions

Authentication – Digital Signature

- **Based on asymmetric keys and digital signature algorithm**
- **Authenticators produced are digital signatures**
- **In a digital-signature algorithm, computationally infeasible to derive $S(k_s)$ from $V(k_v)$**
  - V is a one-way function
  - Thus, k_v is the public key and k_s is the private key
- **Consider the RSA digital-signature algorithm**
  - Similar to the RSA encryption algorithm, but the key use is reversed
  - Digital signature of message $S(k_s)(m) = H(m)^{k_s} \bmod N$
  - The key k_s again is a pair d, N, where N is the product of two large, randomly chosen prime numbers p and q
  - Verification algorithm is $V(k_v)(m, a) \equiv (a^{k_v} \bmod N = H(m))$
    - Where k_v satisfies $k_v k_s \bmod (p - 1)(q - 1) = 1$

Digital Certificates

- **Proof of who or what owns a public key**
- **Public key digitally signed a trusted party**
- **Trusted party receives proof of identification from entity and certifies that public key belongs to entity**
- **Certificate authority are trusted party – their public keys included with web browser distributions**
  - They vouch for other authorities via digitally signing their keys, and so on