Asymmetric Cryptography

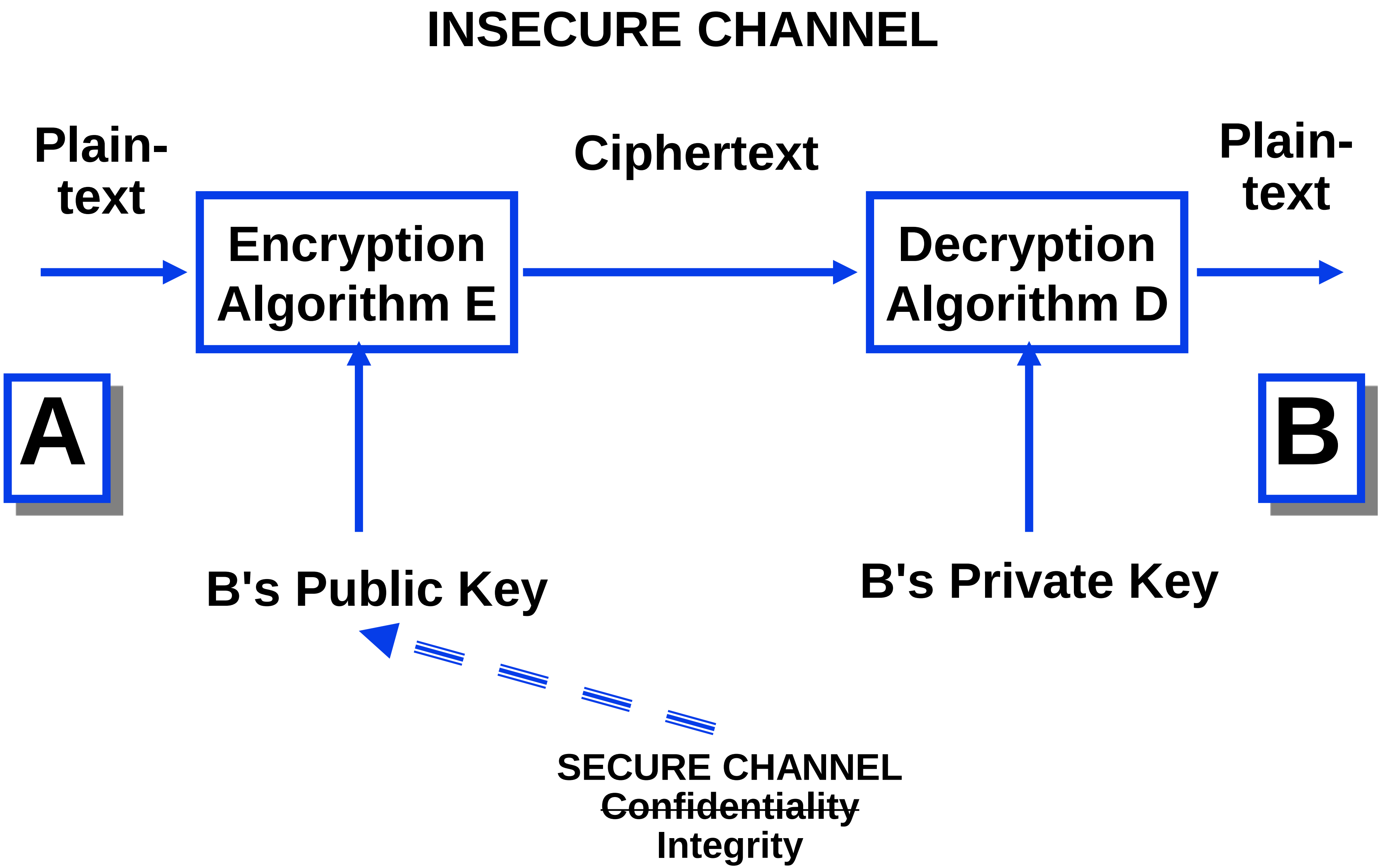
Prof. Ravi Sandhu
Executive Director and Endowed Chair

Lecture 3

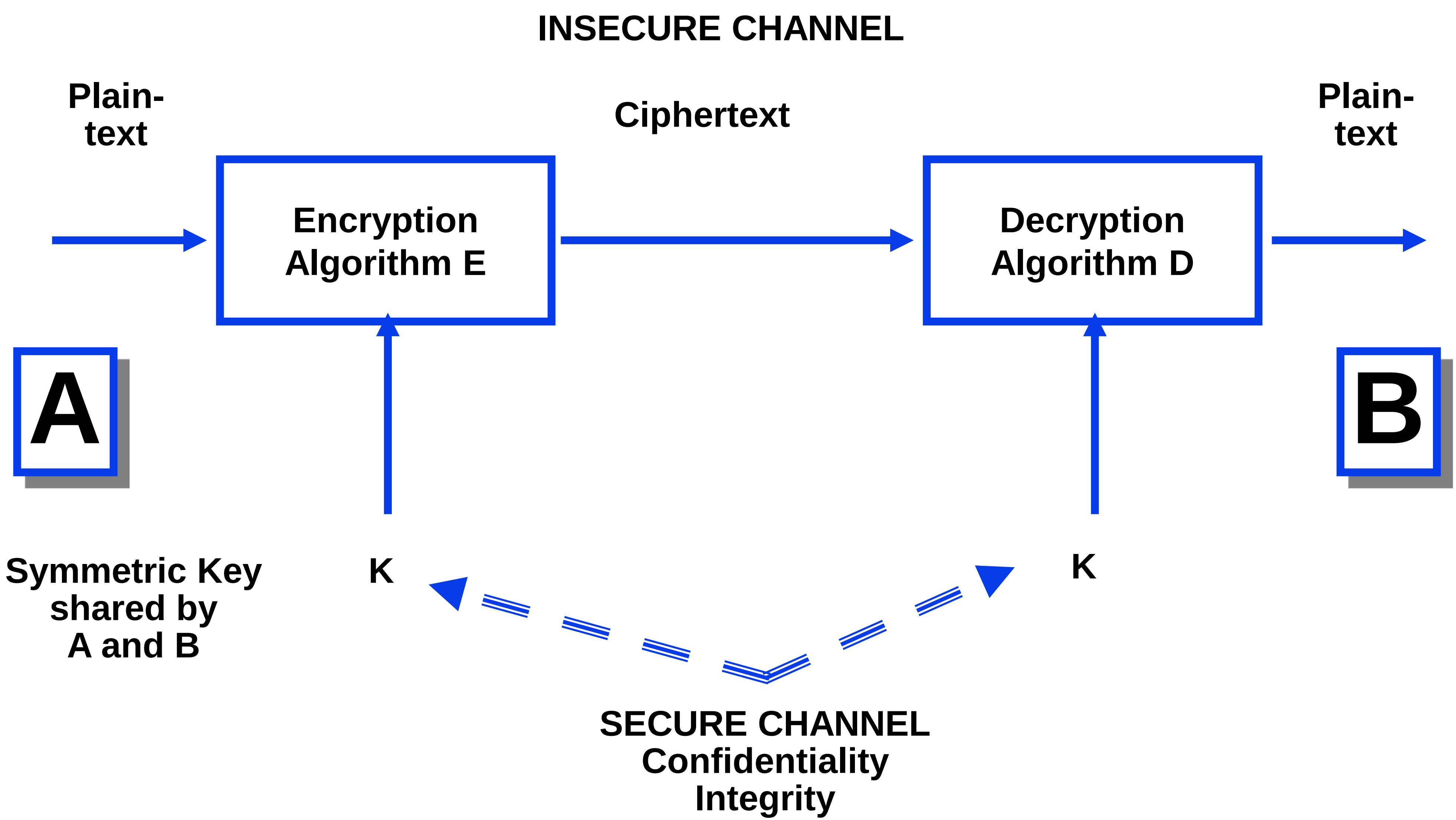
ravi.utsa@gmail.com
www.profsandhu.com

Asymmetric Encryption

Public-Key Encryption



Symmetric-Key Encryption



Public-Key Encryption

- reduces the key distribution problem to a secure channel for authentic communication of public keys
- requires authentic dissemination of 1 public key/party
- scales well for large-scale systems
  - with N parties we need to generate and distribute N public keys

Known Public-Key Attack

- confidentiality based on infeasibility of computing B's private key from B's public key
- key sizes are large (2048 bits and above) to make this computation infeasible

Speed

- public key runs 1000 times slower than symmetric key
  - think 2g versus 4g on smartphone
- This large difference in speed is likely to remain
  - Maybe reduce to 100 times

- Use public keys to distribute symmetric keys, use symmetric keys to protect data

RSA Cryptosystem

- public key is (n,e)
- private key is d
- encrypt: $C = M^e \bmod n$
- decrypt: $M = C^d \bmod n$

X
Not covered in lecture

RSA Cryptosystem

- public key is (n,e)
- private key is d
- encrypt: $C = M^e \bmod n$
- decrypt: $M = C^d \bmod n$

X
Not covered in lecture

This naive use of RSA is not secure but will suffice for our purposes

RSA Key Generation

- choose 2 large prime numbers p and q
- compute n = p * q
- pick e relatively prime to (p-1)*(q-1)
- compute d, e*d = 1 mod (p-1)*(q-1)
- publish (n,e)
- keep d private (and discard p, q)

X
Not covered in lecture

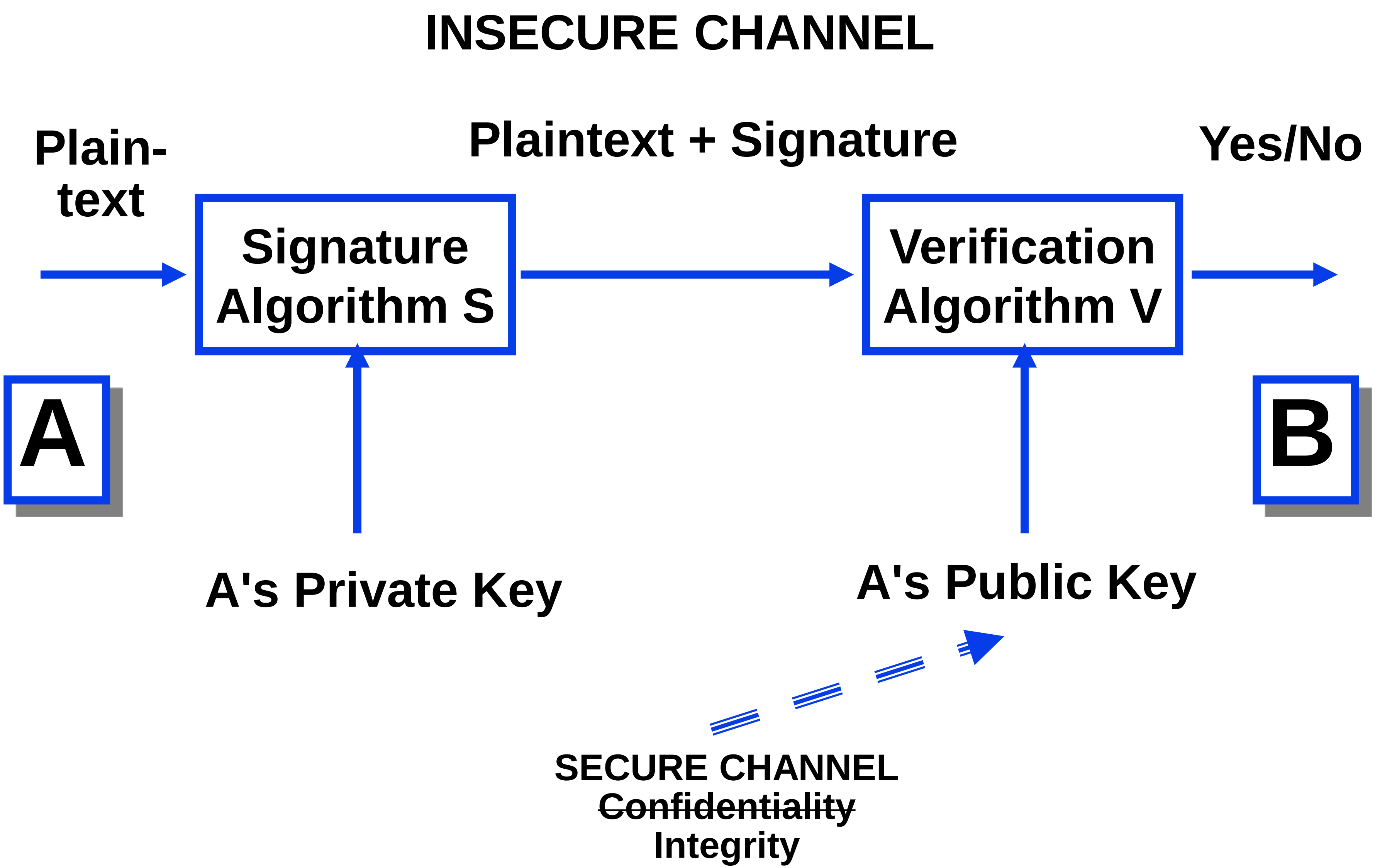
RSA Key Protection

- compute d, e*d = 1 mod (p-1)*(q-1)
- if factorization of n into p*q is known, this is easy to do
- security of RSA is no better than the difficulty of factoring n into p, q

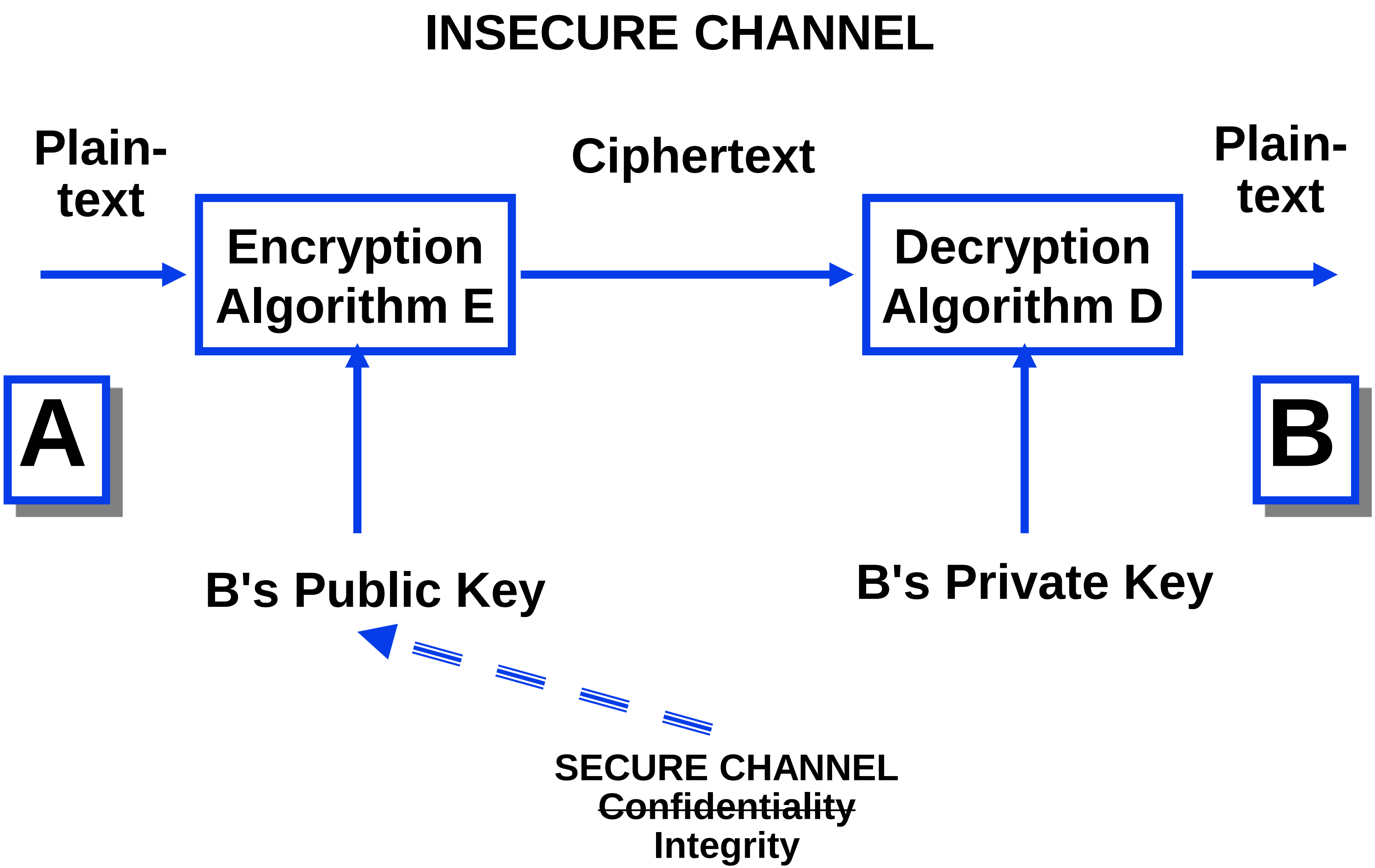
X
Not covered in lecture

Asymmetric Digital Signatures

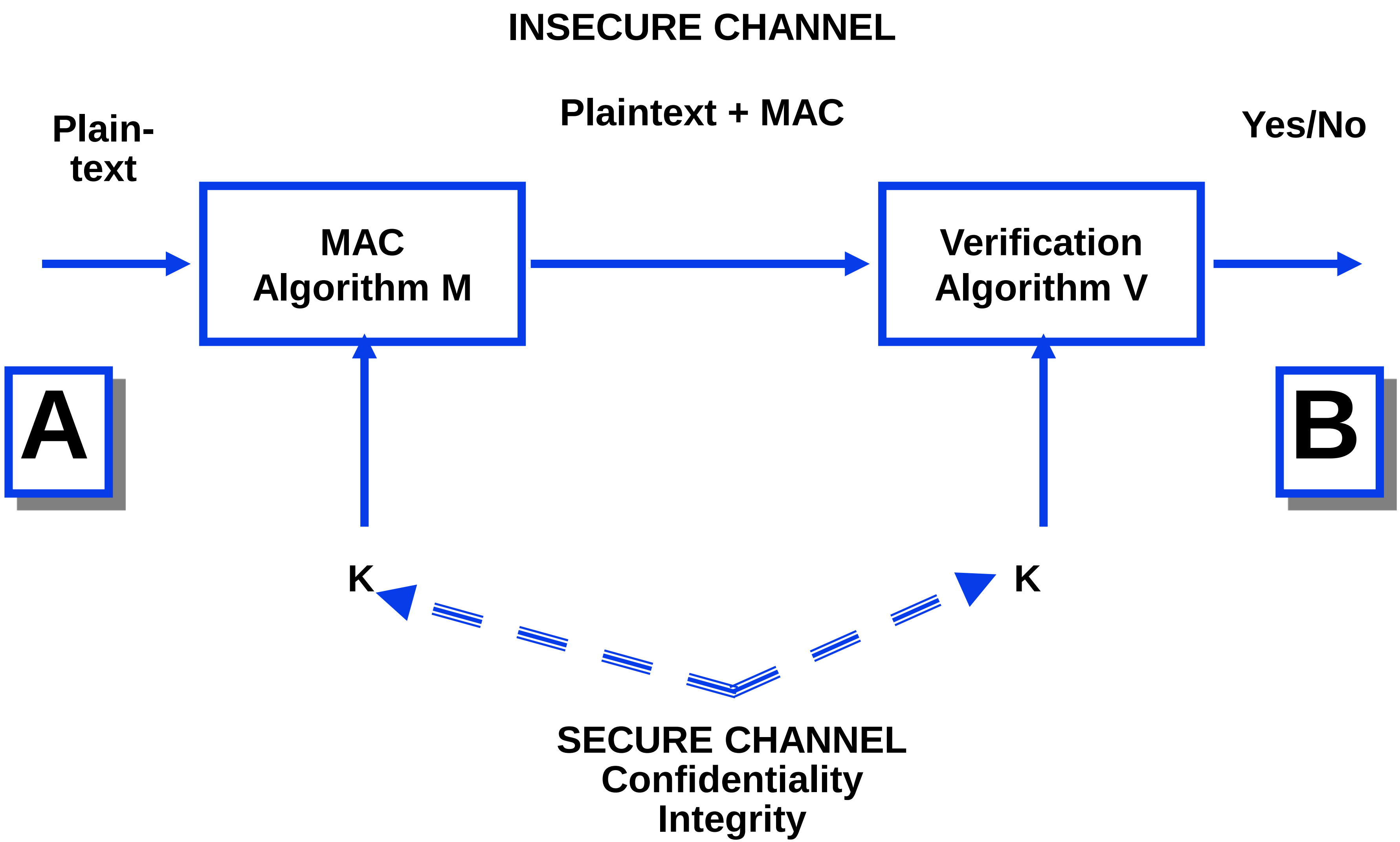
Public-Key Digital Signature



Compare Public-Key Encryption



Compare Symmetric Key MAC



Digital Signatures in RSA

- RSA has a unique property, not shared by other public key systems
- Encryption and decryption commute
- $(M^e \bmod n)^d \bmod n = M$ encryption
- $(M^d \bmod n)^e \bmod n = M$ signature
- Same public key can be use for encryption and signature
  - But not recommended

X
Not covered in lecture

Message Digest

Encryption Speed Revisited

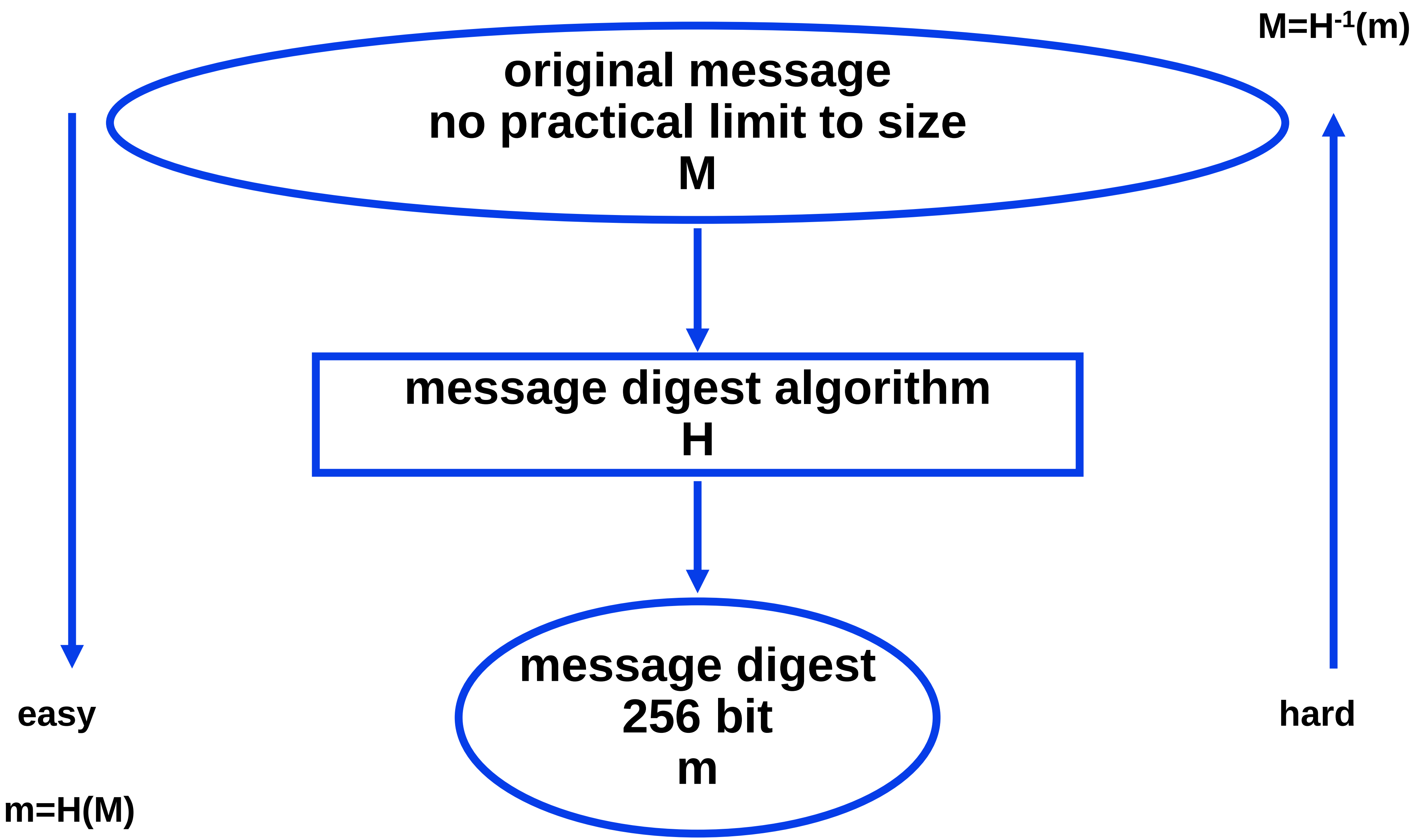
- public key runs 1000 times slower than symmetric key
  - think 2g versus 4g on smartphone
- This large difference in speed is likely to remain
  - Maybe reduce to 100 times

- Use public keys to distribute symmetric keys, use symmetric keys to protect data

Digital Signature Speed

- public key runs 1000 times slower than symmetric key
  - think 2g versus 4g on smartphone
- This large difference in speed is likely to remain
  - Maybe reduce to 100 times
- Sign the message digest (or hash) not the message

Message Digest (Hash)



World-Leading Research with Real-World Impact!

Desired Characteristics

- weak hash function
  - difficult to find M' such that H(M')=H(M)
- given M, m=H(M) try messages at random to find M’ with H(M’)=m
  - 2^k trials on average, k=128 to be safe

Desired Characteristics

- strong hash function
  - difficult to find any two M and M' such that H(M')=H(M)
- try pairs of messages at random to find M and M' such that H(M')=H(M)
  - $2^{k/2}$ trials on average, k=256 to be safe

Birthday paradox

Message Authentication Code

Symmetric Encryption Based

CBC-MAC

MAC has same size as block size of underlying cryptosystem

CCM mode
Provides confidentiality and integrity

Message-Digest Based

HMAC

Hash the message and a symmetric key

MAC has same size as underlying hash function or can truncate

Revisiting after discussing message digests

Asymmetric Key Exchange

Diffie-Hellman Key Agreement

A

$y_A = a^{x_A} \bmod p$

public key

$y_B = a^{x_B} \bmod p$

public key

B

private key

x_A

private key

x_B

$$k = y_B^{x_A} \bmod p = y_A^{x_B} \bmod p = a^{x_A * x_B} \bmod p$$

system constants: p: prime number, a: integer

X
Not covered in lecture

Diffie-Hellman Key Agreement

- security depends on difficulty of computing x given $y=a^x \bmod p$
- called the discrete logarithm problem

X
Not covered in lecture

Diffie-Hellman Man-in-the-Middle Attack

A

C

B

X
Not covered in lecture

Public keys need to be authenticated