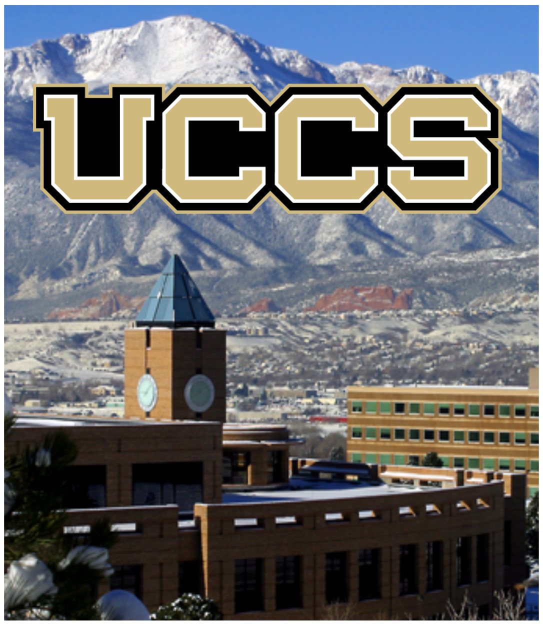


Asymmetric Cryptography and Key Management

Asymmetric Cryptography Overview

Sang-Yoon Chang, Ph.D.

Module: Asymmetric Cryptography Overview

Principles and Misconceptions

Framework and Key Use

Cipher Requirements and
Trapdoor One-Way Function

Revisiting Symmetric Cryptography

Uses one key shared by Alice and Bob

Security relies on the key secrecy

Also called private-key or secret-key cryptography

Asymmetric Cryptography

There is a public key and a private key

Also called public-key cryptography

Asymmetric since Alice and Bob are
not equal

Misconceptions of Asymmetric Cryptography

Asymmetric cryptography is more secure than symmetric cryptography

Asymmetric cryptography replaces symmetric cryptography

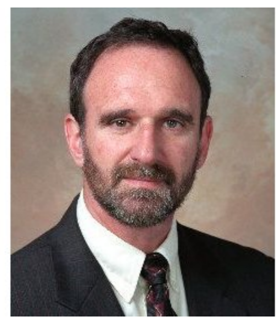
Key distribution is trivial

Asymmetric Cryptography Invention

Invented to address two issues:

- Key distribution
- Digital signatures

Diffie and Hellman,
1976

Asymmetric Cryptography

A pair of key, one of which is public and the other private/secret

Alice uses one key and Bob the other

Infeasible to derive the private key from the public key or the ciphertext

Asymmetric Cipher for Different Security Uses

Symmetric cipher for confidentiality

Asymmetric cipher for confidentiality
or authentication, depending on the
key use and the cipher design

Cryptography Terminology

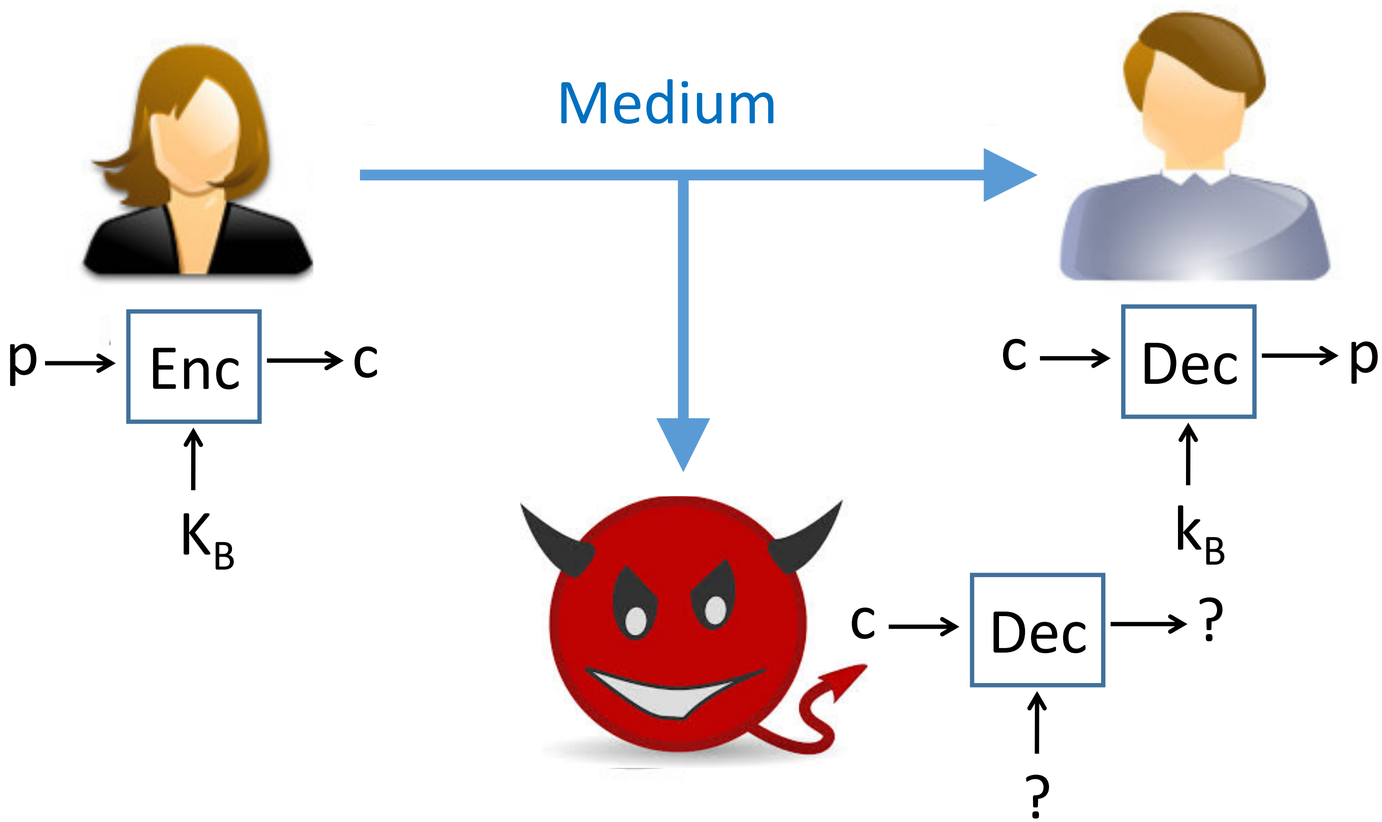
Plaintext (p) - the original message
Ciphertext (c) - the coded message
Private Key (k_i) - User i’s private key
Public Key (K_i) - associated with user i and paired with k_i

Cryptography Terminology

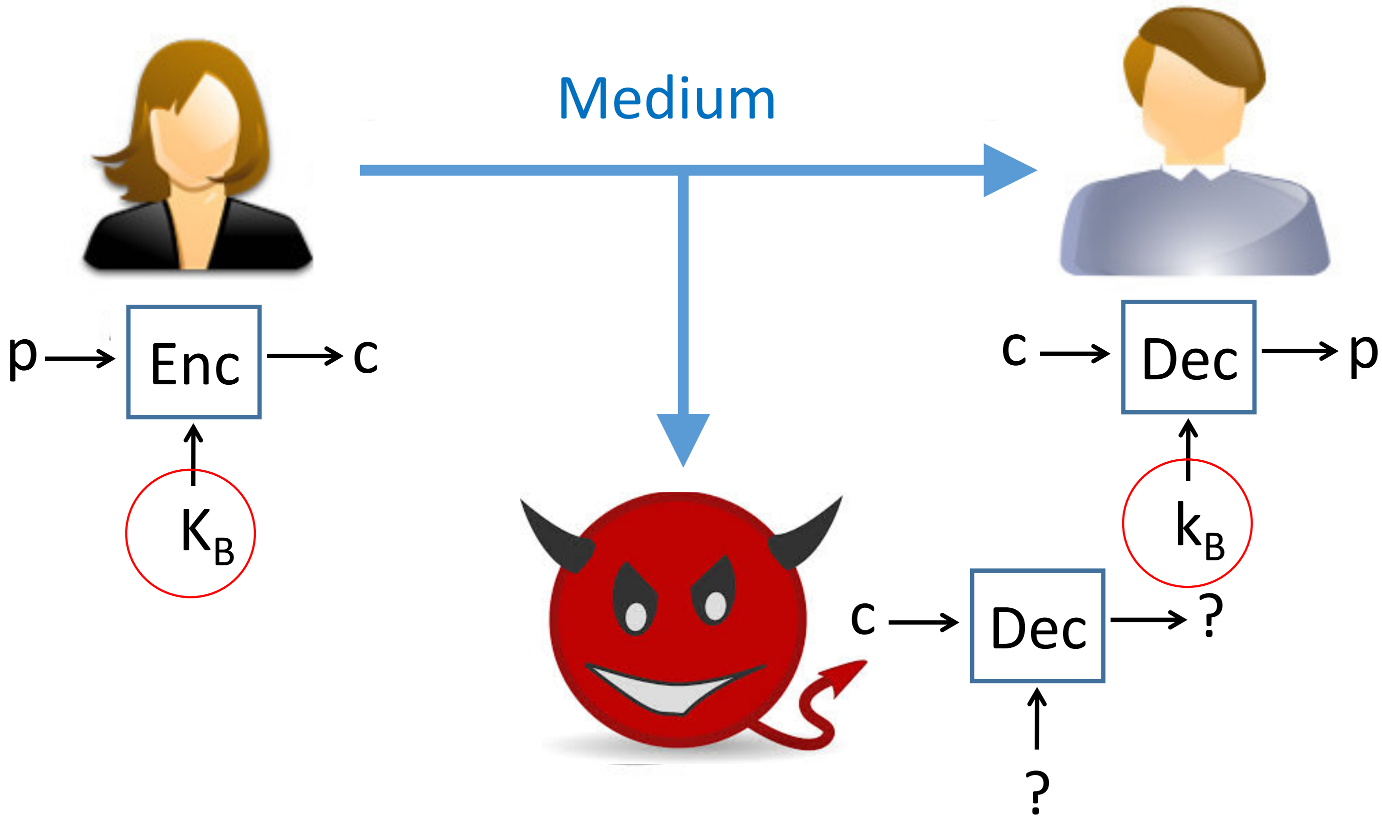
Plaintext (p) - the original message
Ciphertext (c) - the coded message
Private Key (k_i) - User i's private key
Public Key (K_i) - associated with user i and paired with k_i

For any p,
$Dec_1(k_i,Enc_1(K_i,p))=p$ for confidentiality
$Dec_2(K_i,Enc_2(k_i,p))=p$ for authentication

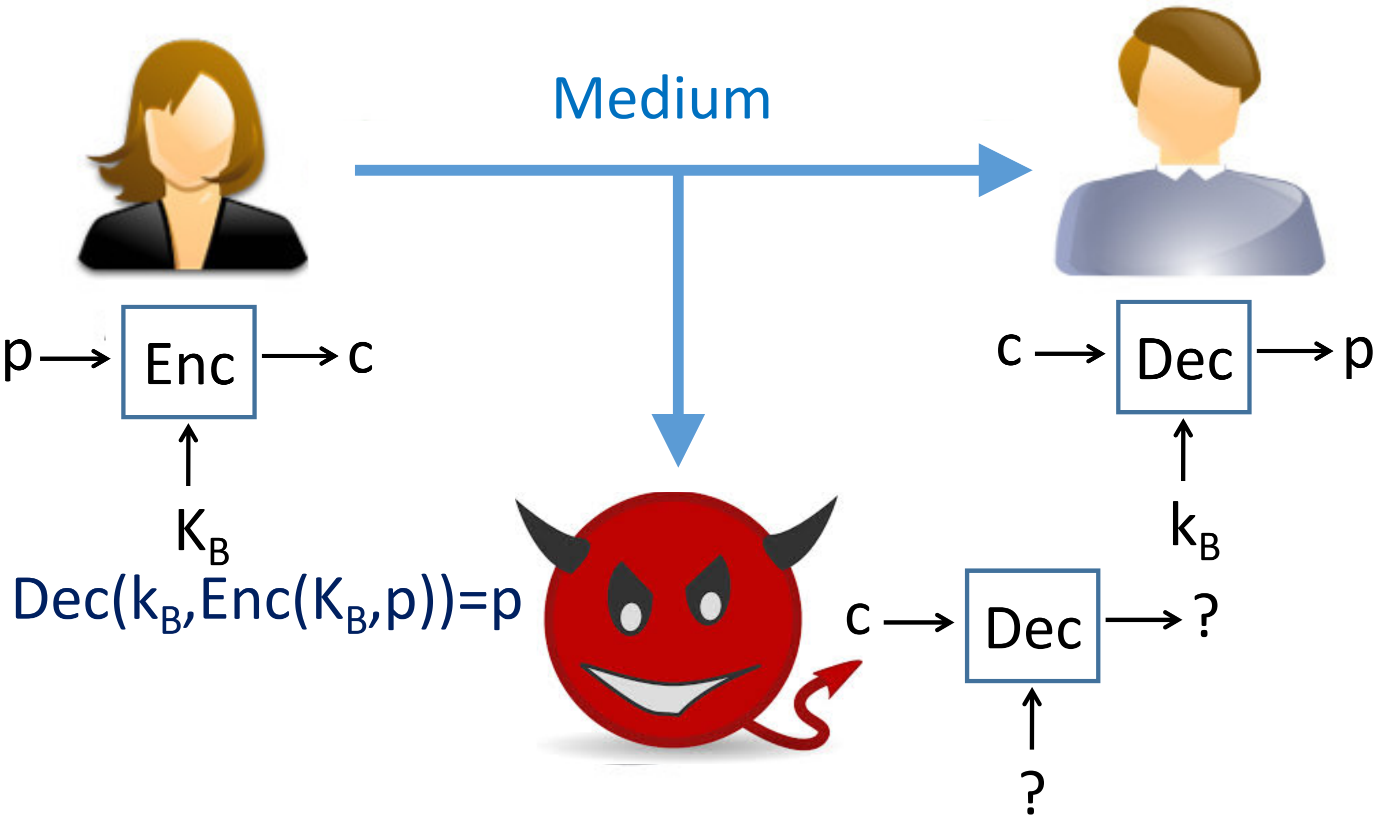
Asymmetric: Alice uses Bob’s public key K_B



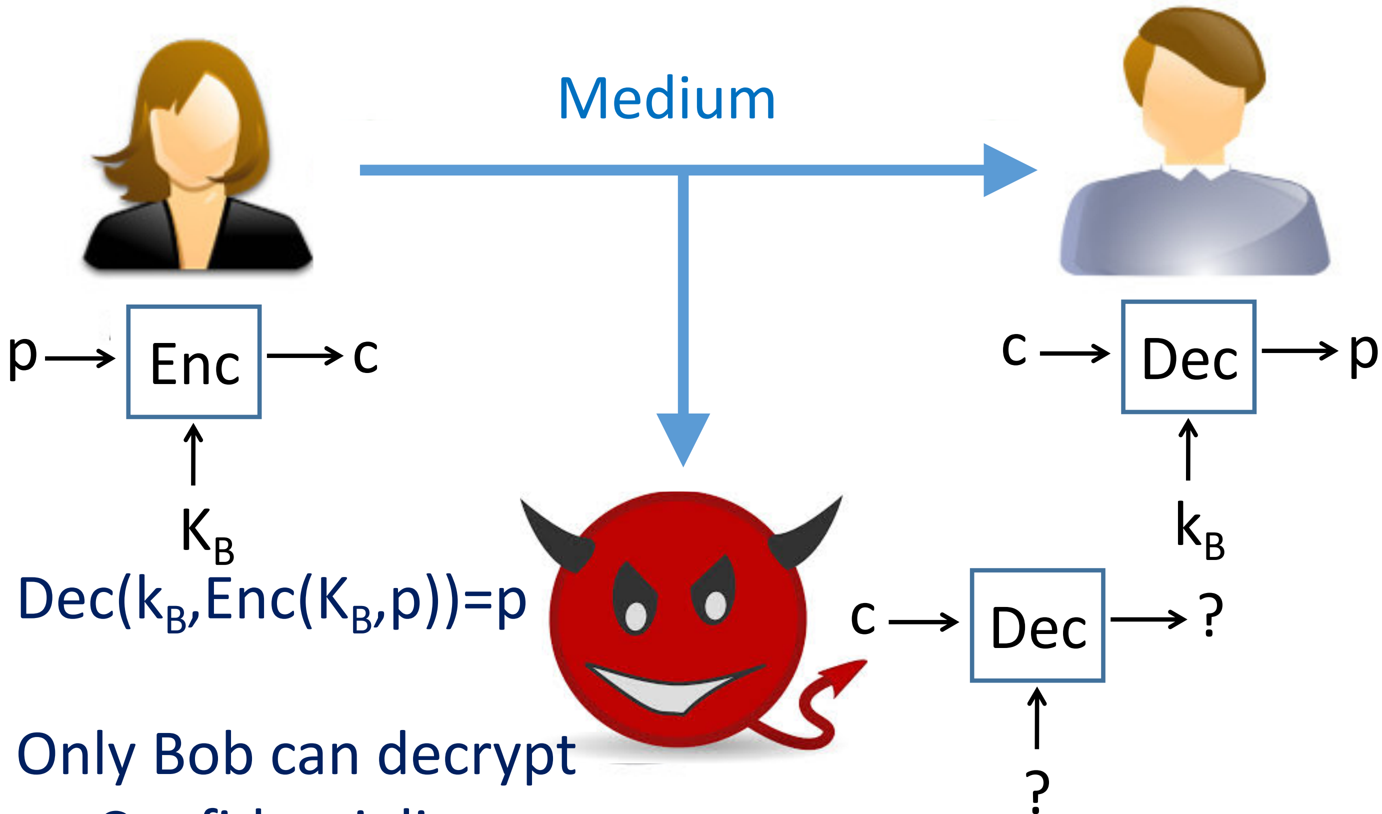
Asymmetric: Alice uses Bob's public key K_B



Asymmetric: Alice uses Bob's public key K_B



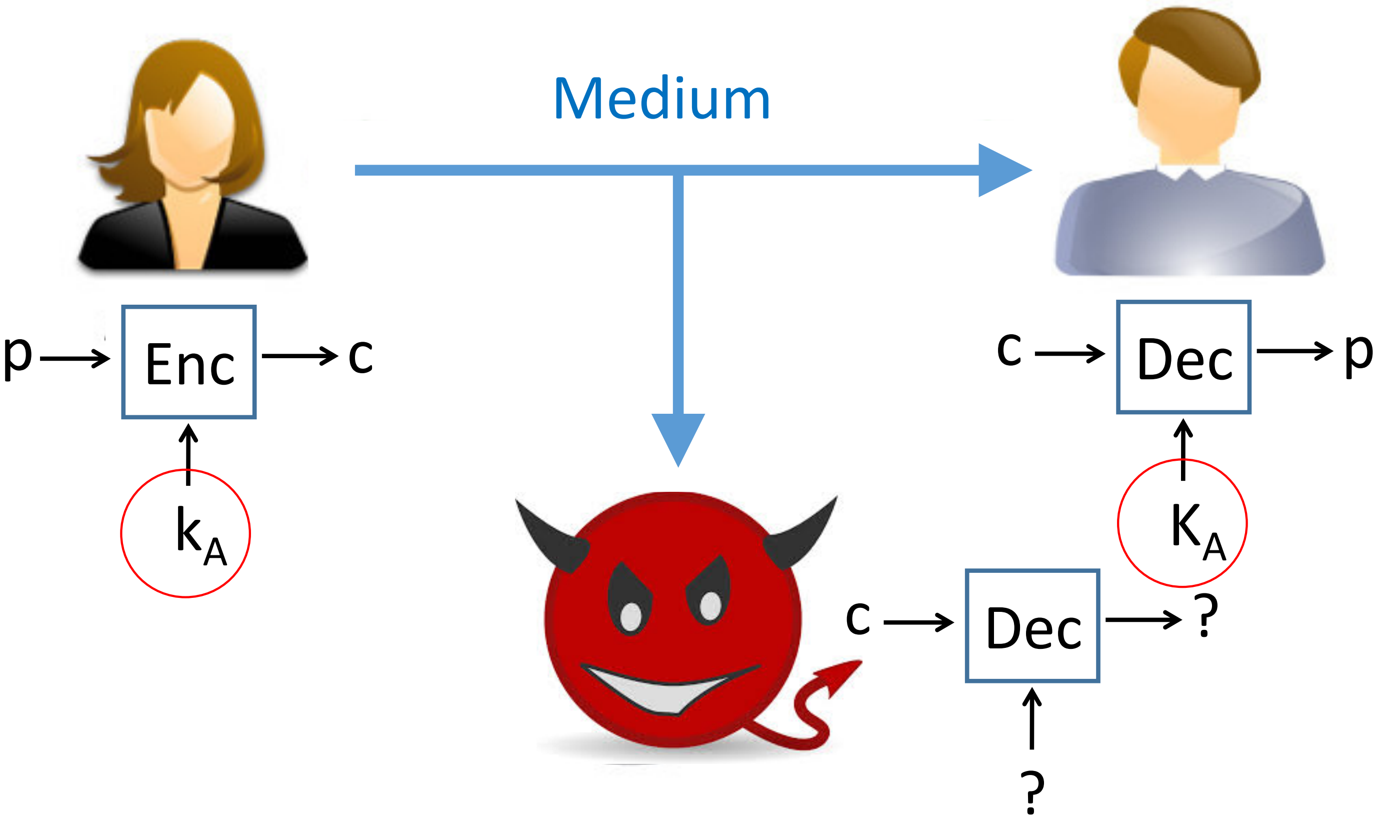
Asymmetric: Alice uses Bob's public key K_B



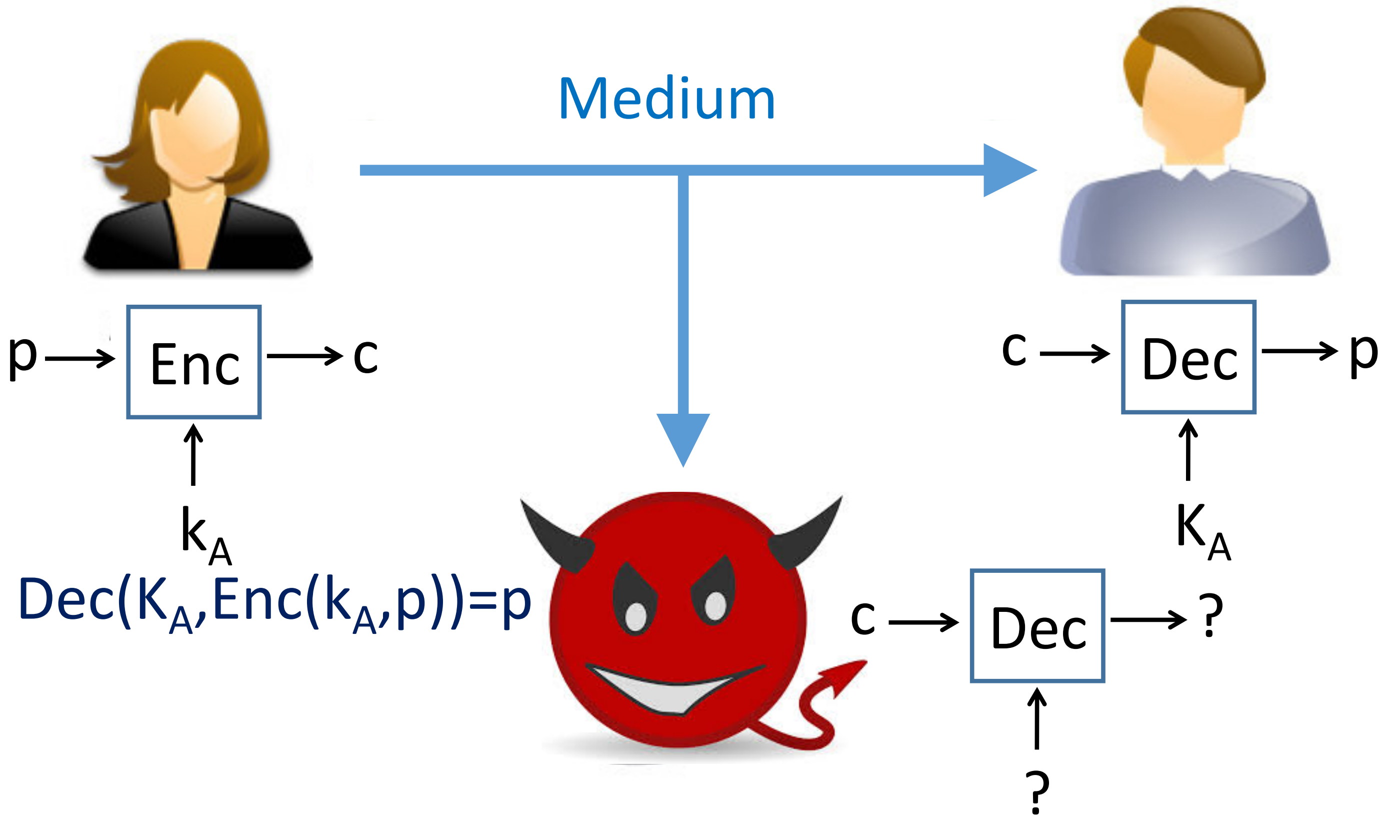
$Dec(k_B,Enc(K_B,p))=p$

Only Bob can decrypt
=> Confidentiality

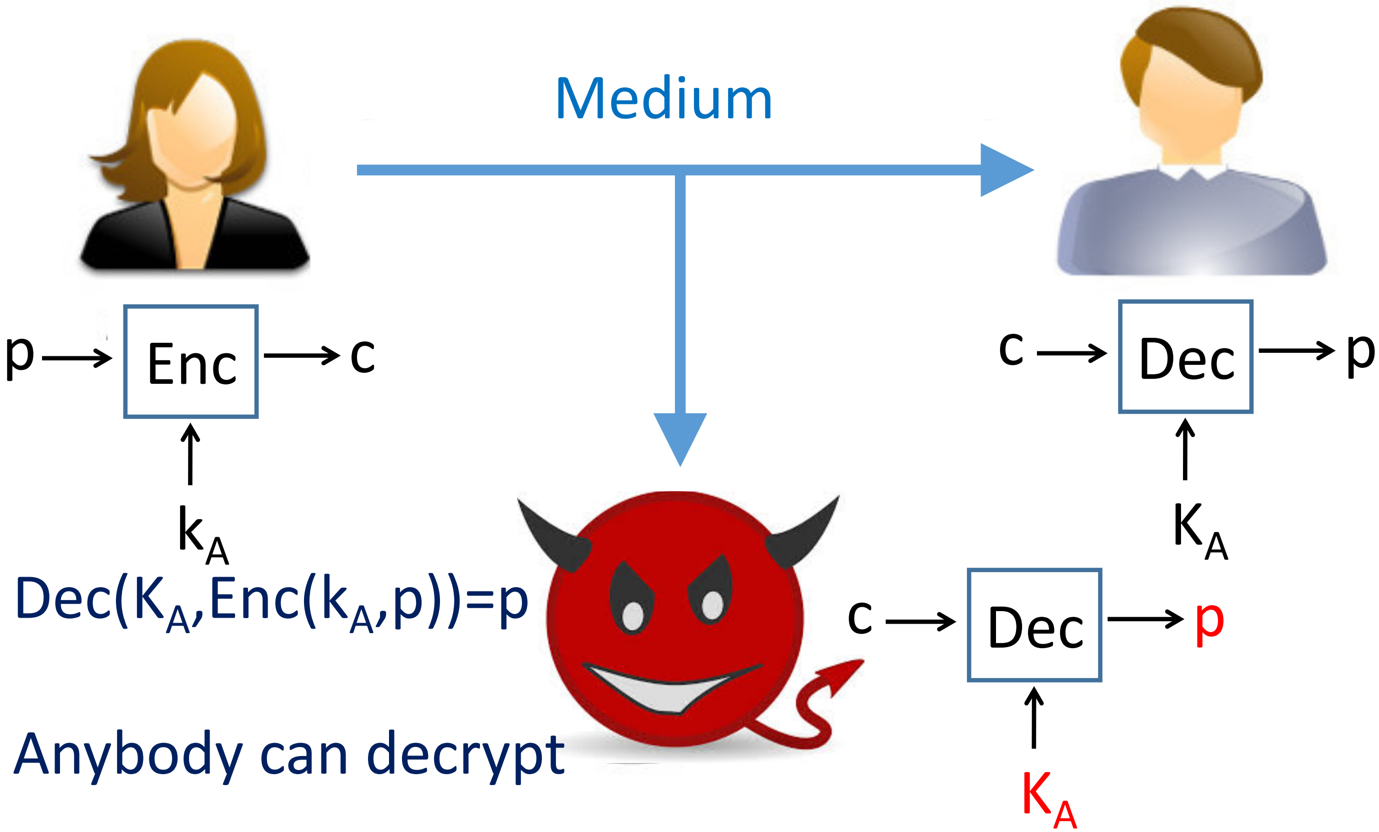
Asymmetric: Alice uses Alice's private key k_A



Asymmetric: Alice uses Alice's private key k_A



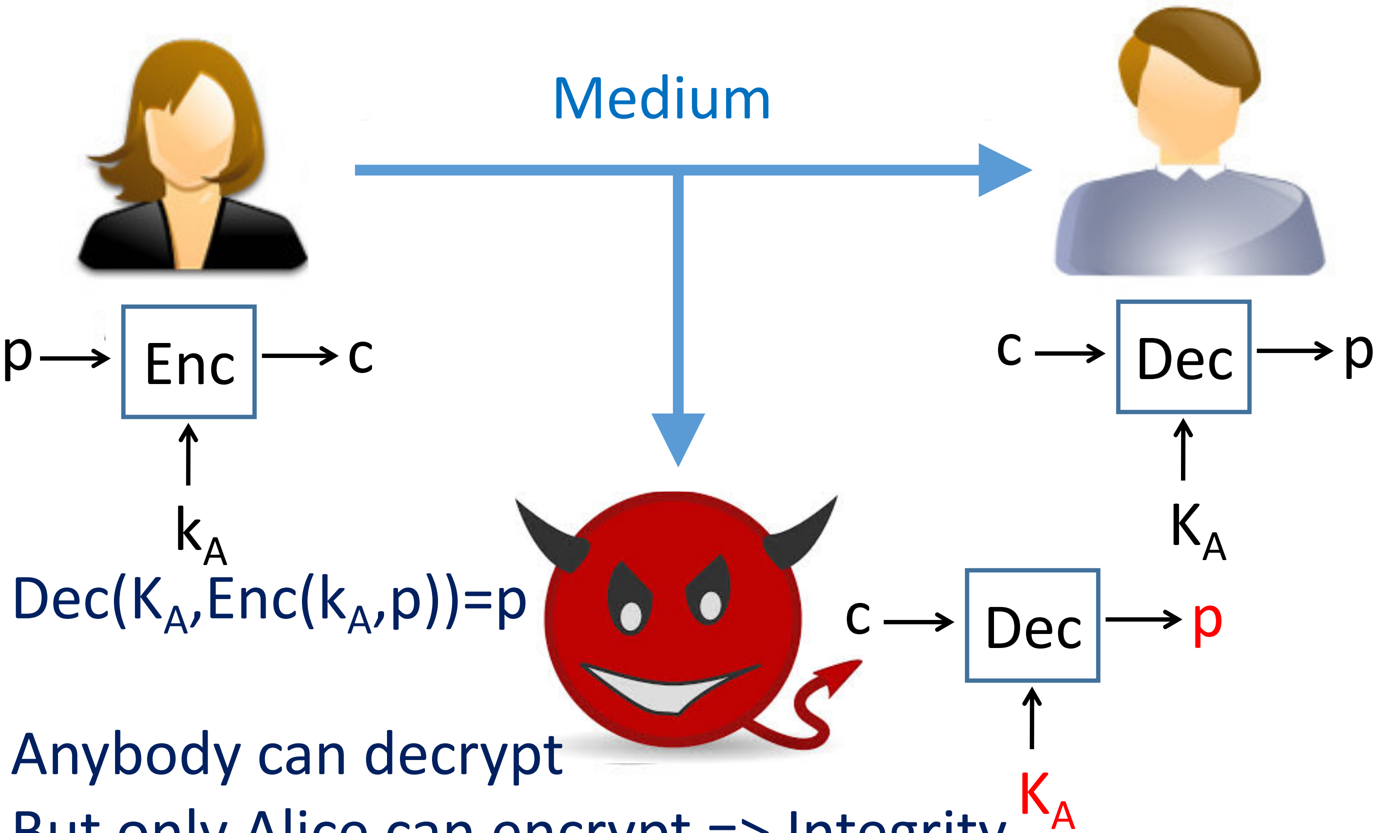
Asymmetric: Alice uses Alice's private key k_A



Dec(K_A,Enc(k_A,p))=p

Anybody can decrypt

Asymmetric: Alice uses Alice's private key k_A



$Dec(K_A, Enc(k_A, p)) = p$

Anybody can decrypt

But only Alice can encrypt => Integrity

Asymmetric Cryptography Applications

Alice uses K_B and Bob uses k_B
=> Confidentiality protection on p

Alice uses k_A and Bob uses K_A
=> Authentication and source integrity

Asymmetric Cryptography Applications

Alice uses K_B and Bob uses k_B

=> Confidentiality protection on p

Encryption (confidentiality)

Alice uses k_A and Bob uses K_A

=> Authentication and source integrity

Digital signature

Key exchange

Asymmetric Cryptography Applications

Alice uses K_B and Bob uses k_B

=> Confidentiality protection on p

Encryption (confidentiality) RSA

Alice uses k_A and Bob uses K_A

=> Authentication and source integrity

Digital signature RSA

Key exchange RSA

Asymmetric Cryptography Applications

Alice uses K_B and Bob uses k_B

=> Confidentiality protection on p

Encryption (confidentiality) RSA

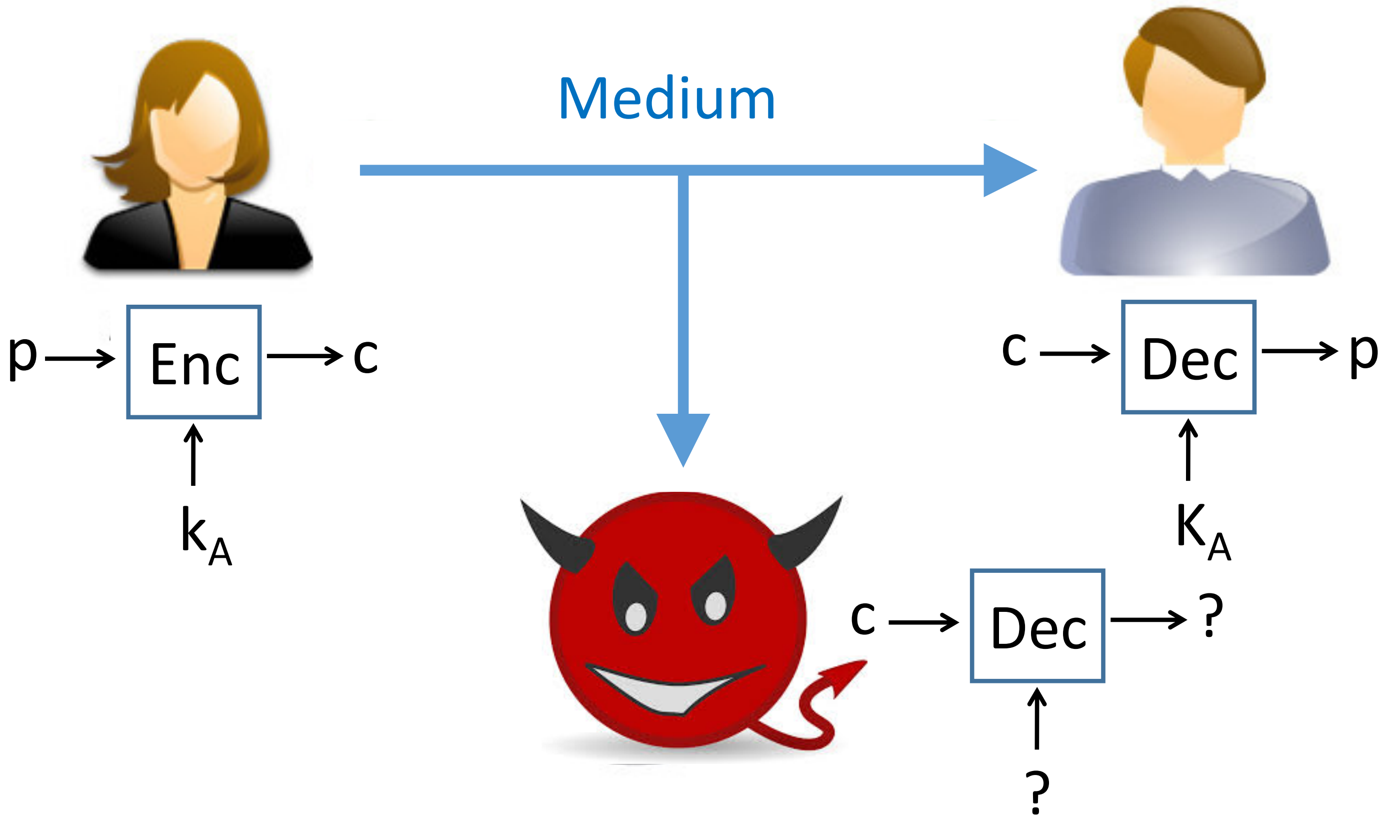
Alice uses k_A and Bob uses K_A

=> Authentication and source integrity

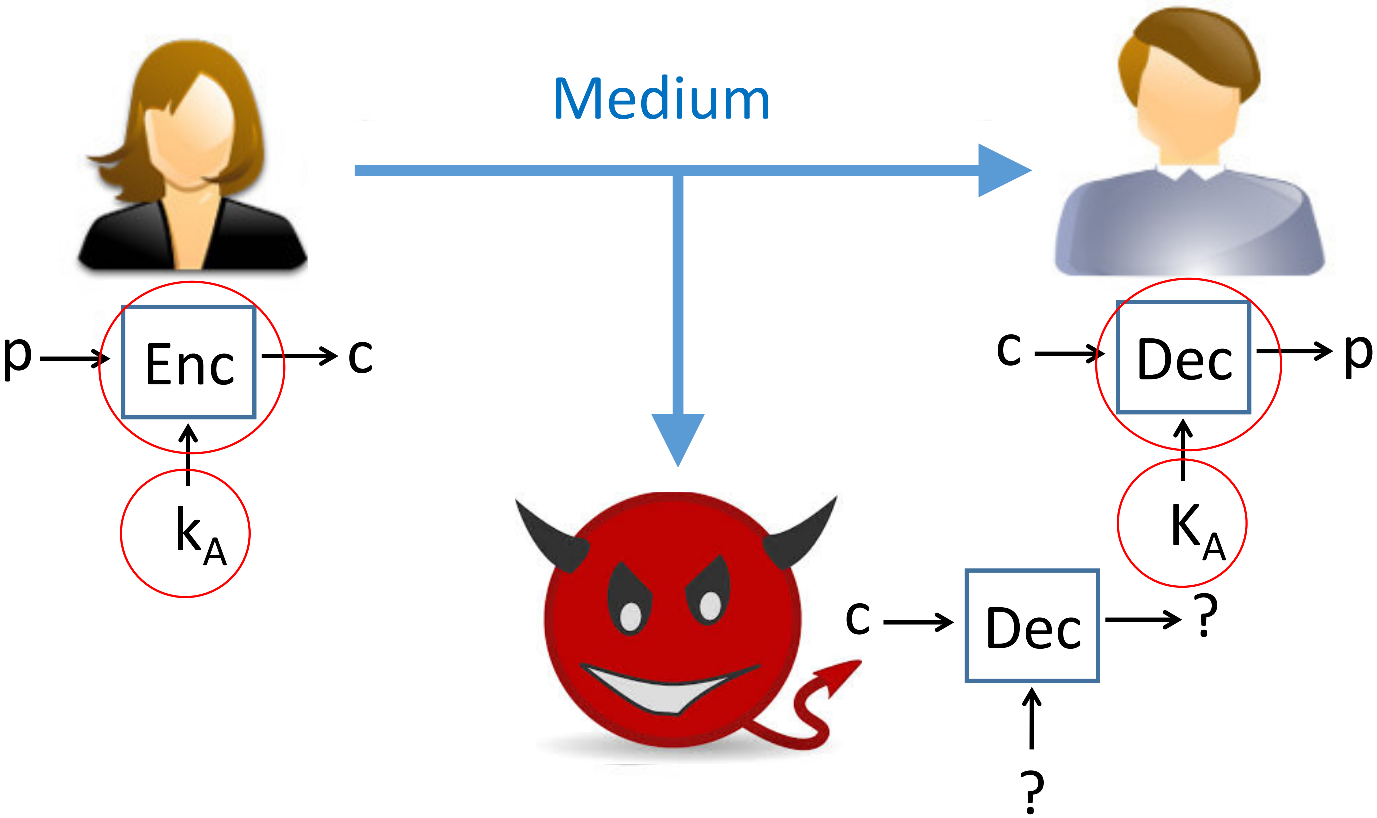
Digital signature RSA DSS

Key exchange RSA D.H.-key-exchange

Asymmetric: Alice uses Alice's private key k_A



Asymmetric: Alice uses Alice's private key k_A



Asymmetric Cipher Requirements

- It is computationally easy for any user i to generate the key pair (k_i,K_i)
- Enc., and Dec. computations are easy

Asymmetric Cipher Requirements

- It is computationally easy for any user i to generate the key pair (k_i,K_i)
- Enc., and Dec. computations are easy

- It is computationally infeasible for Eve to derive k_i from K_i

- It is computationally infeasible for Eve to derive p from K_i and C

Asymmetric Cipher Requirements

- It is computationally easy for any user i to generate the key pair (k_i,K_i)
- Enc., and Dec. computations are easy

- It is computationally infeasible for Eve to derive k_i from K_i

- It is computationally infeasible for Eve to derive p from K_i and C
- (Optional) Keys can be in both order:
  p =Dec(K_i,Enc(k_i,p))=Dec(k_i,Enc(K_i,p))
  E.g., RSA

Trapdoor One-Way Function

One-way function is:

$y=f(x)$ is easy; $x=f^{-1}(y)$ is infeasible

Trapdoor one-way function is:

- $y = f_k(x)$ easy, if k and x are known
- $x = f_k^{-1}(y)$ easy, if k and y are known
- $x = f_k^{-1}(y)$ infeasible, if y known but k unknown