Asymmetric Ciphers I

Computer and Network Security

Emilio Coppa

Symmetric ciphers

Alice

x → $E_K(x)$ → y → $D_K(y)$ → x Bob

K K
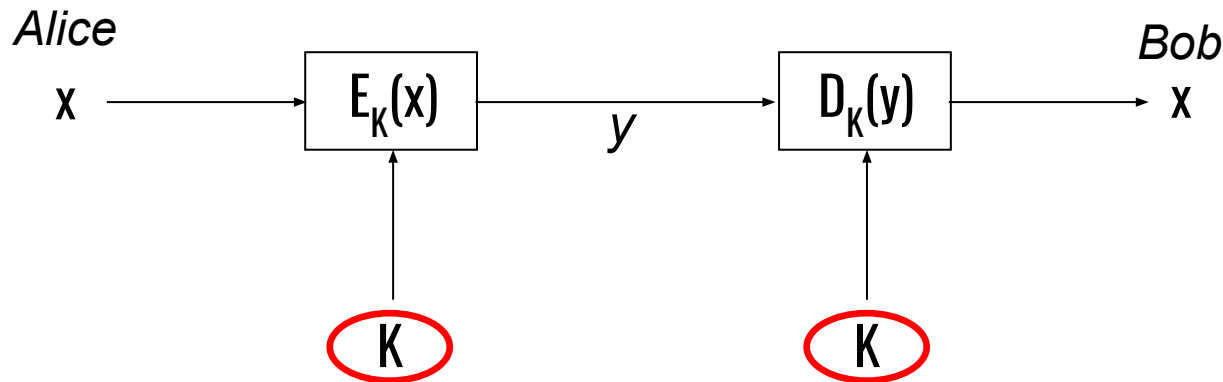
Two properties of symmetric (secret-key) crypto-systems:

- The same secret key K is used for encryption and decryption
- Encryption and Decryption are very similar (or even identical) functions

Symmetric ciphers: shortcomings

1. **Key Distribution Problem**: the secret key must be transported securely.... How?

2. **Number of Keys**: if n users in a network then

   ○ we need $\dfrac{n \cdot (n-1)}{2}$ keys.

   ○ each user has to store n-1 keys, which is reasonable

   ○ adding one user to the network, we have to transfer securely n keys

3. **Non repudiation:** no protection against cheating by Alice or Bob

Chapter 6 of Understanding Cryptography by Christof Paar and Jan Pelzl

Idea behind Asymmetric Cryptography

New Idea:

Use the "good old mailbox" principle:
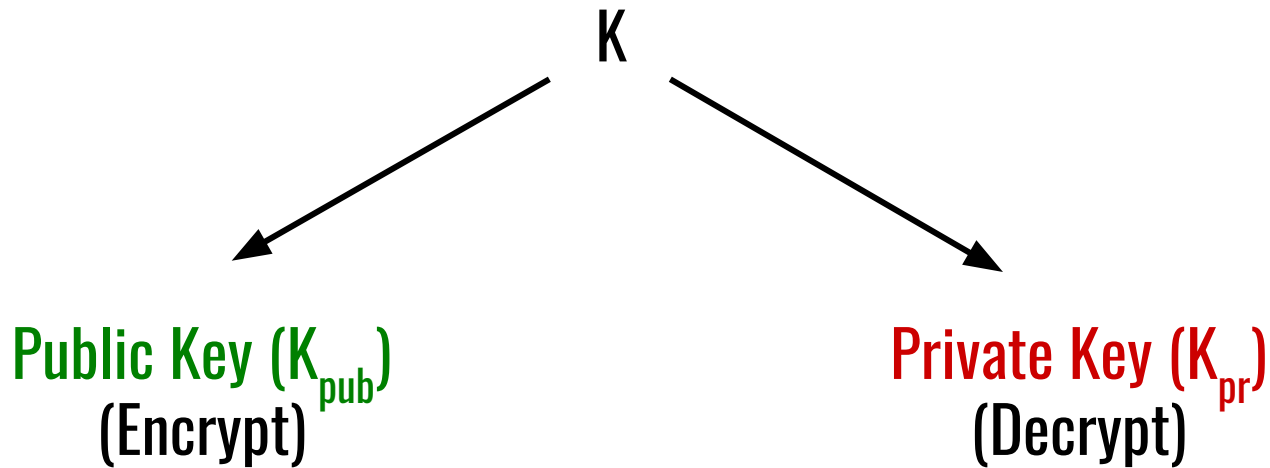
Everyone can drop a letter

But only the owner has the correct key to open the box

1976: first publication of such an algorithm by Whitfield Diffie and Martin Hellman, and also by Ralph Merkle.
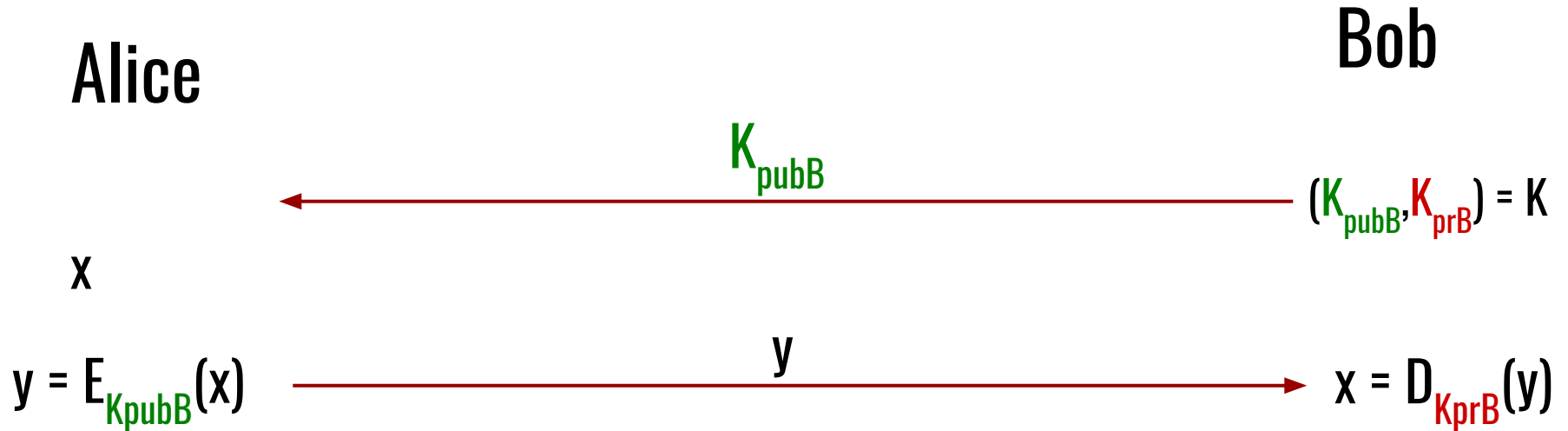
Asymmetric (Public-Key) Cryptography

Principle: "split up" the key

$$K$$

Public Key (K_{pub})
(Encrypt)

Private Key (K_{pr})
(Decrypt)

☐ During the key generation, a key pair K_{pub} and K_{pr} is computed

Basic Protocol for Public-Key Encryption

Alice

Bob

K_{pubB}

$(K_{pubB}, K_{prB}) = K$

x

$y = E_{KpubB}(x)$ ———— y ————→ $x = D_{KprB}(y)$

☐ Key Distribution Problem solved (not entirely true as we need to authenticate the public key)

Security Mechanisms of Public-Key Cryptography

Here are main mechanisms that can be realized with asymmetric cryptography:

- Key Distribution (e.g., Diffie-Hellman key exchange, RSA) without a pre-shared secret (key)

- Nonrepudiation and Digital Signatures (e.g., RSA, DSA or ECDSA) to provide message integrity

- Identification, using challenge-response protocols with digital signatures

- Encryption (e.g., RSA / Elgamal)
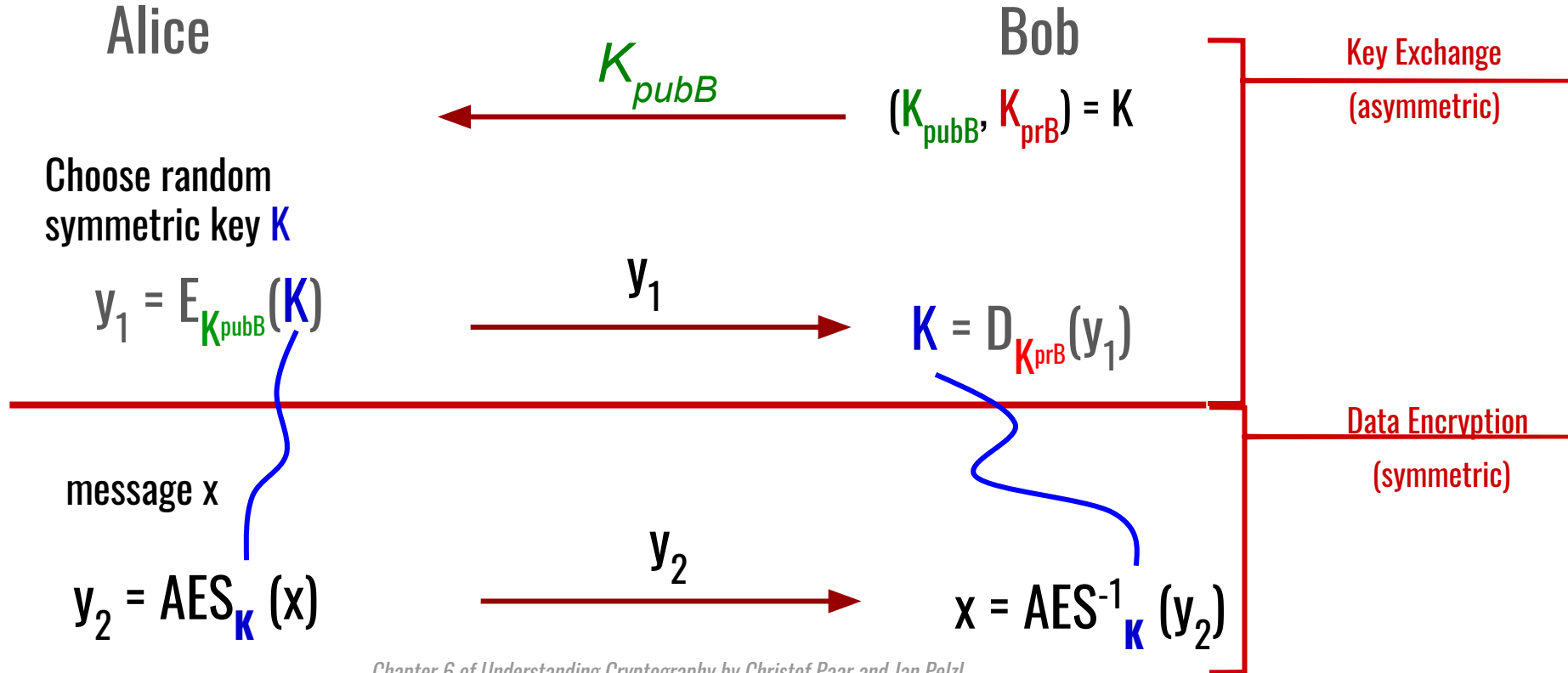 Disadvantage: Computationally very intensive (1000 times slower than symmetric Algorithms!)

Hybrid System

In practice: hybrid systems, incorporating asymmetric and symmetric algorithms

1. Key exchange (for symmetric schemes) and digital signatures are performed with (slow) asymmetric algorithms

2. Encryption of data is done using (fast) symmetric ciphers, e.g., block ciphers or stream ciphers
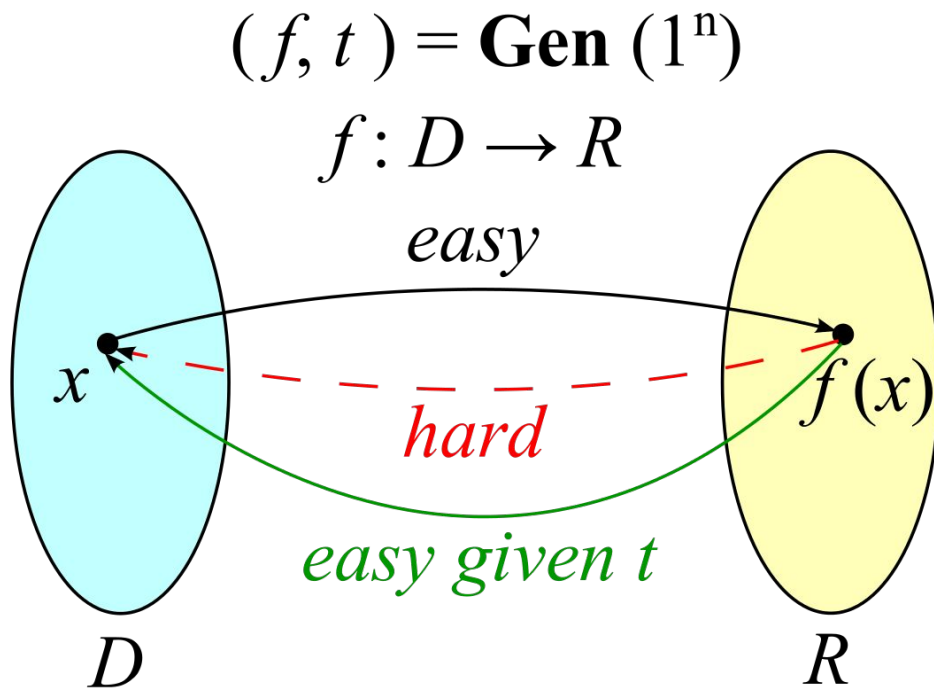
Hybrid System (2)

Example: Hybrid protocol with AES as the symmetric cipher

Alice   Bob

$$K_{pubB}$$ ←

$$(K_{pubB}, K_{prB}) = K$$

Choose random symmetric key K

$$y_1 = E_{K_{pubB}}(K)$$

$$y_1 \rightarrow$$

$$K = D_{K_{prB}}(y_1)$$

message x

$$y_2 = AES_K(x)$$

$$y_2 \rightarrow$$

$$x = AES^{-1}_K(y_2)$$

Key Exchange (asymmetric)

Data Encryption (symmetric)

How to build PK schemes?

One idea is to use a One Way Function (OWF):

$$(f, t) = \mathbf{Gen}(1^n)$$

$$f: D \rightarrow R$$

easy

hard

easy given t

x

D

$f(x)$

R

- Computing f(x) is easy
- Computing x given f(x) is hard
- Computing x given f(x) and t is easy

t is called a trapdoor

Existence of One Way Functions

There is no proof that OWFs actually exist (the proof would imply that P != NP).
However, there are a few good candidates (no one proved yet they are not one way):

- **[IF] integer factorization** with prime numbers:
  $f(x) = p * q$ where p and q are prime numbers is easy to compute
  given $f(x)$ is hard to perform factorization to get p and q

- **[DL] discrete logarithm**:
  $f(x) = a^b \bmod p$ where is p is prime is easy to compute
  given $f(x)$ is hard to compute $b = \log_a f(x)$

- **[ECC] Elliptic Curves**: based on elliptic curve discrete logarithm problem

Key Lengths and Security Levels

Symmetric	ECC	RSA, DL	Remark
64 Bit	128 Bit	≈ 700 Bit	Only short term security (a few hours or days)
80 Bit	160 Bit	≈ 1024 Bit	Medium security (except attacks from big governmental institutions etc.)
128 Bit	256 Bit	≈ 3072 Bit	Long term security (without quantum computers)

ECC are more recent than RSA/DL and thus are less used in practice although they seem to be very powerful.

Credits

These slides are based on material from:

- Slides of Prof. D'Amore from CNS 2019-2020

- Christof Paar and Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer. http://www.crypto-textbook.com/

- Wikipedia (english version)