Public-Key Encryption

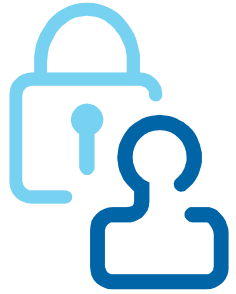
Security Engineering, Week 3

Dr Arman Khouzani, Dr Na Yao

Learning outcomes

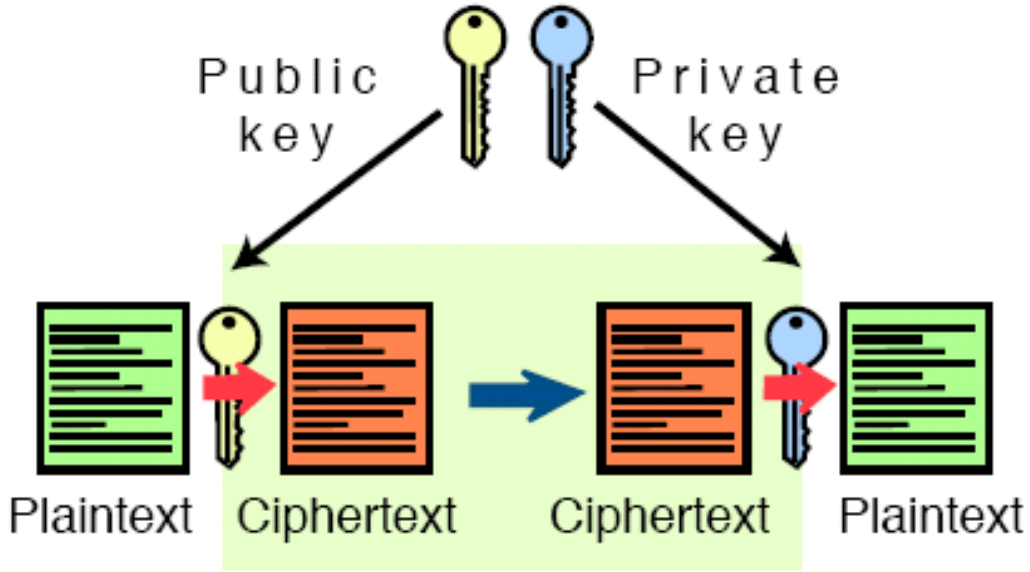
- Explain the basic principles behind public-key cryptography.
- Recognise the fundamental problems which need to be solved before public-key cryptography can be used effectively.
- Recognise the security mechanism behind RSA and ElGamal.
- Identify the main uses of public-key cryptography.

Public-Key Encryption



- **So far, we looked at 'Symmetric Encryption'**
  - Encryption and decryption use the same key

- Problem: Key negotiation processes (key establishment) …

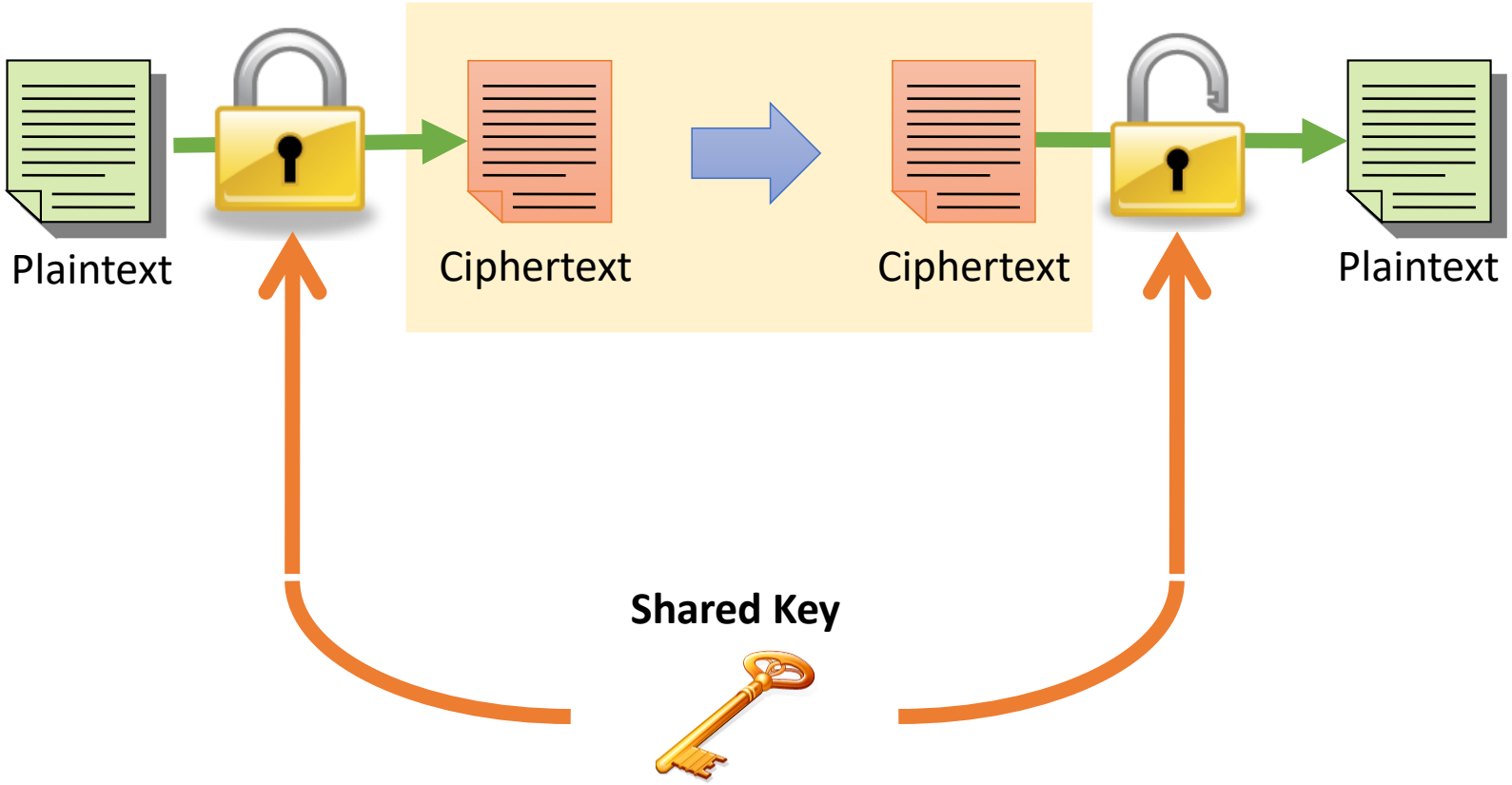


Overview

- **Symmetric (conventional) encryption**
  - Encryption and decryption use the same key
- **Asymmetric encryption**
  - Encryption and decryption use different keys:
    - public key
    - Private key
- **Public-Key is an ‘Asymmetric’ method**
  - Uses different keys
  - Based on mathematical functions (instead of permutation and substitution methods)
  - It should provide enhanced features:
    - Confidentiality.
    - Key distribution.
    - Authentication,

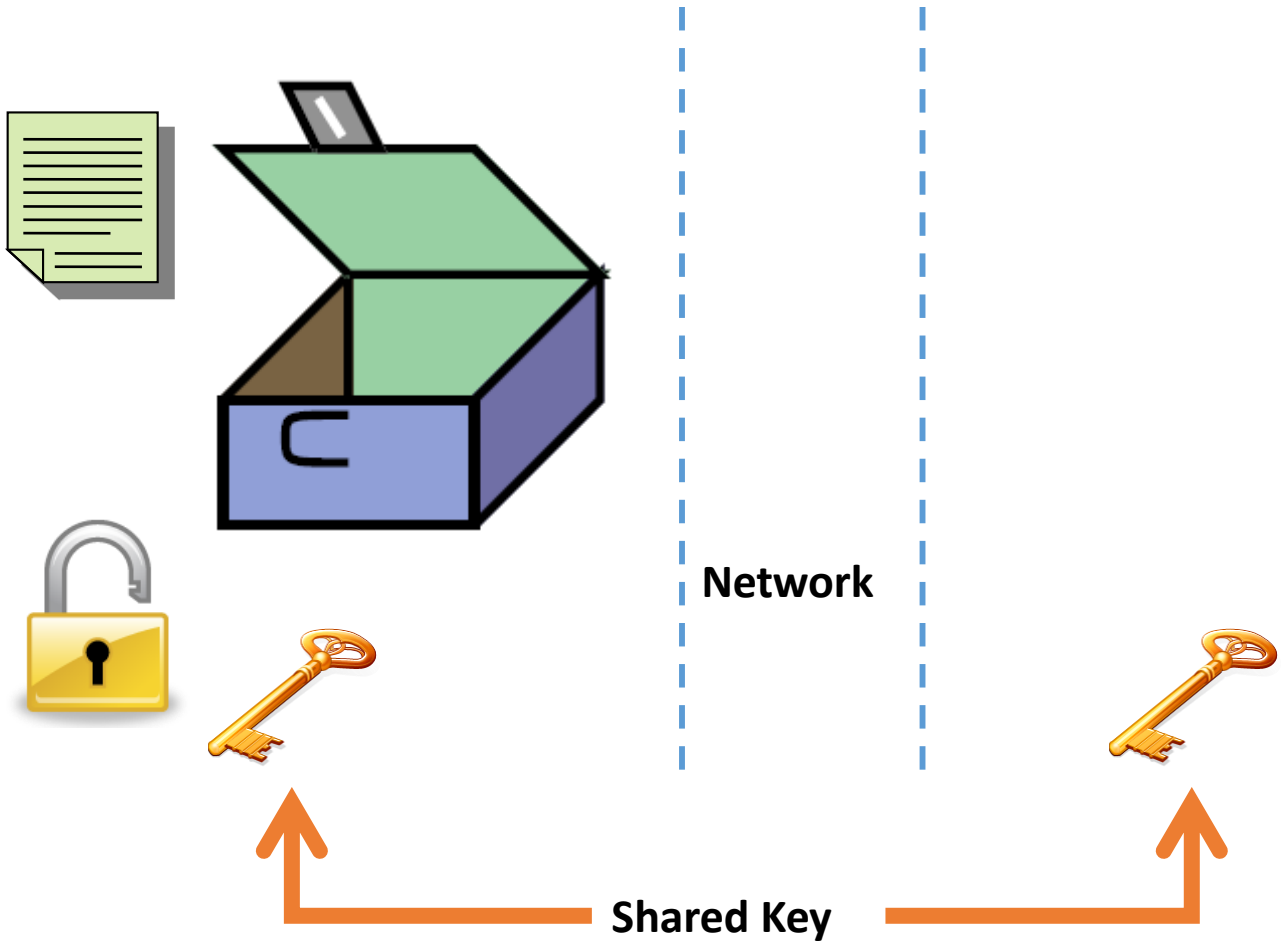
Symmetric Encryption

- Encryption and decryption share the same key



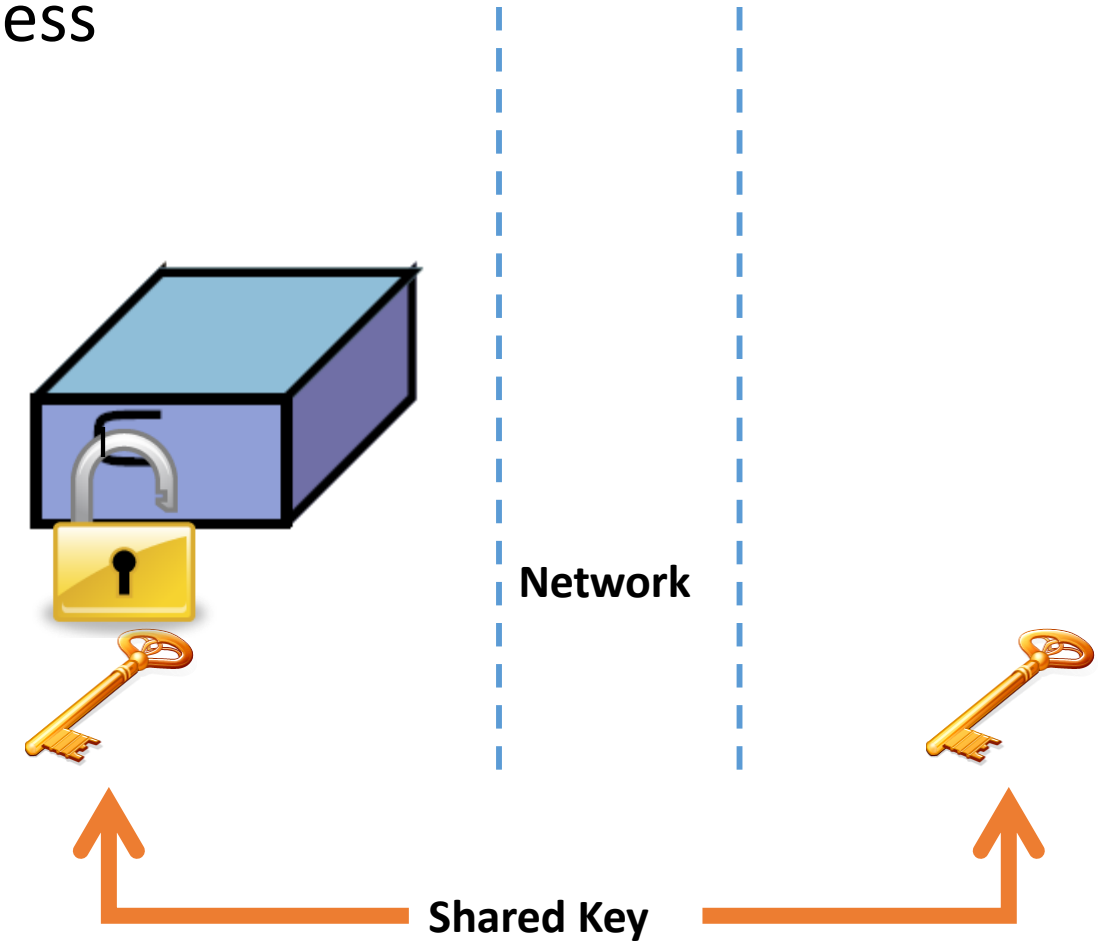
Symmetric Encryption

- Plaintext



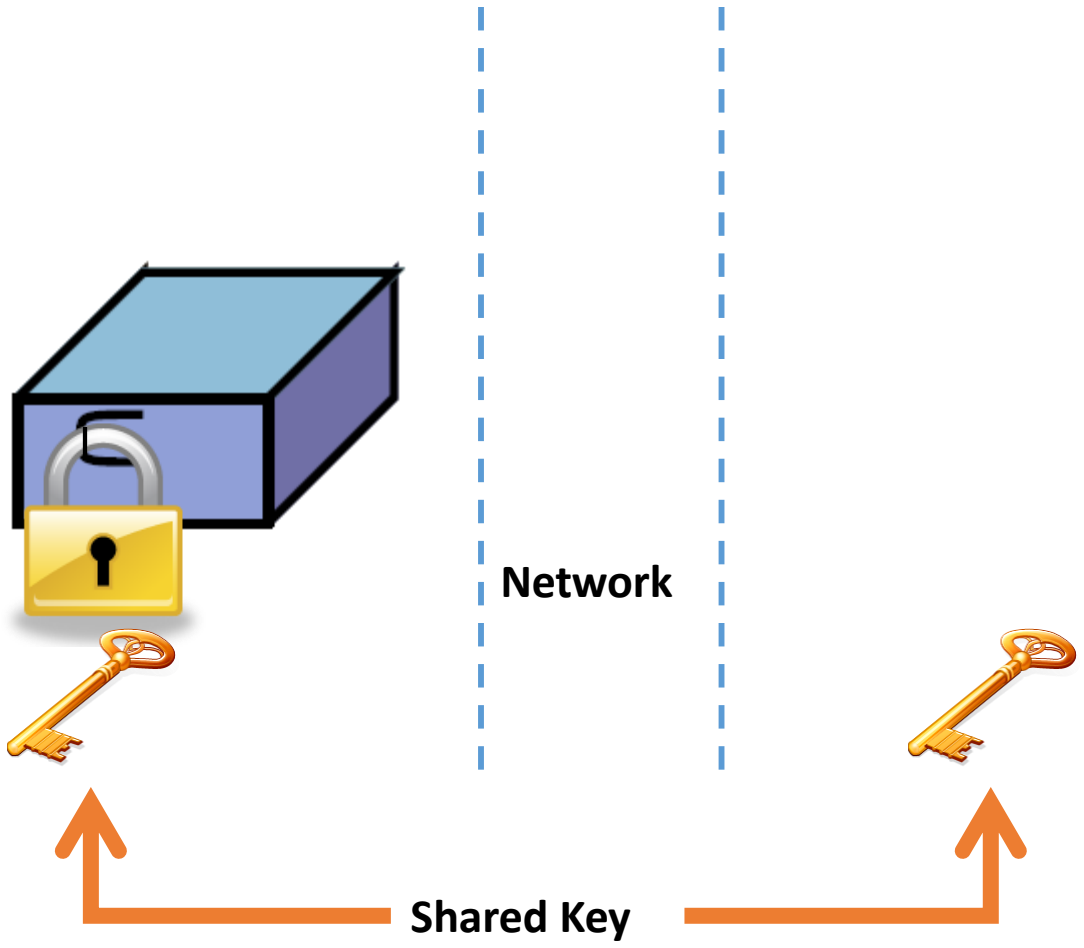
Symmetric Encryption

- Encryption process



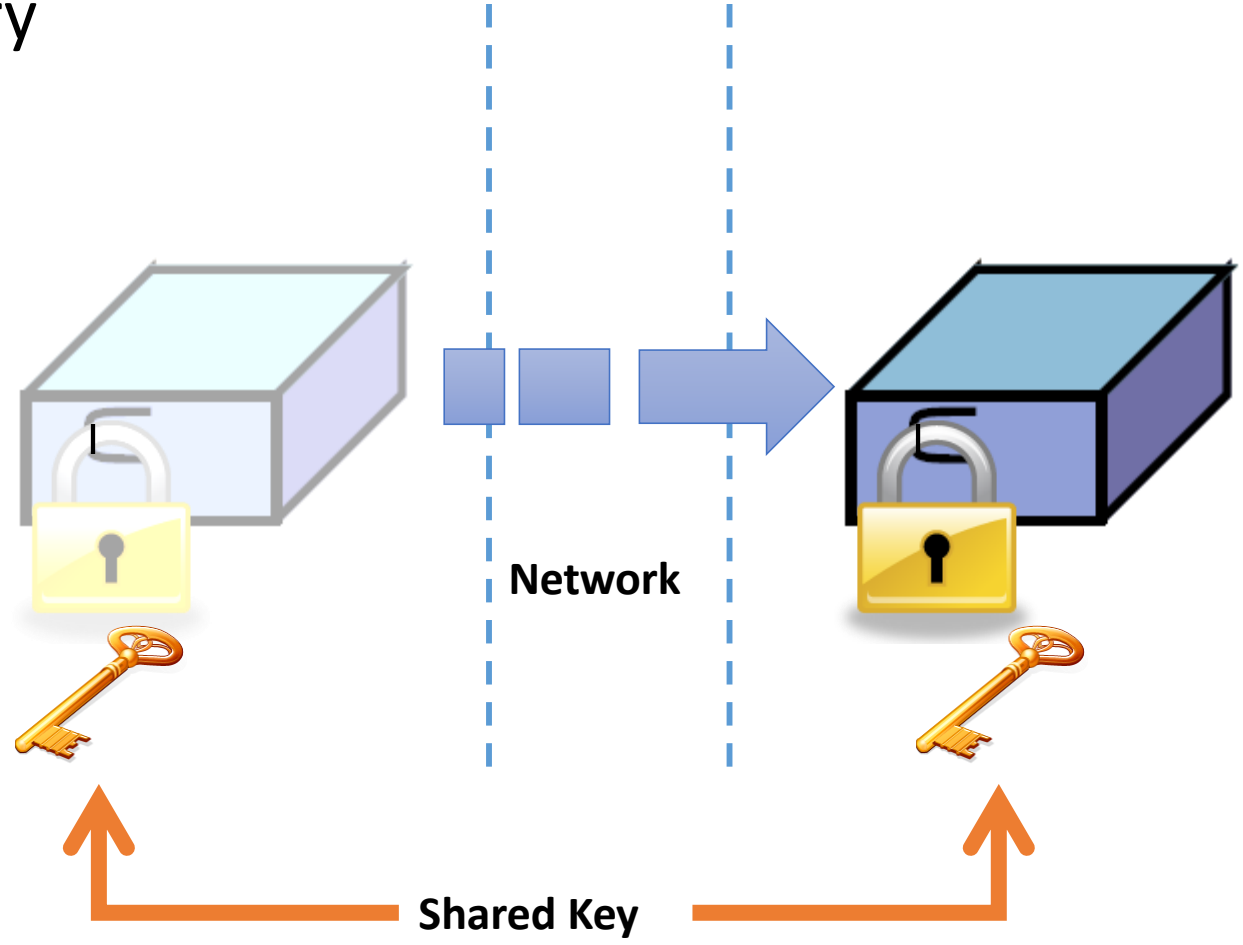
Symmetric Encryption

- Ciphertext



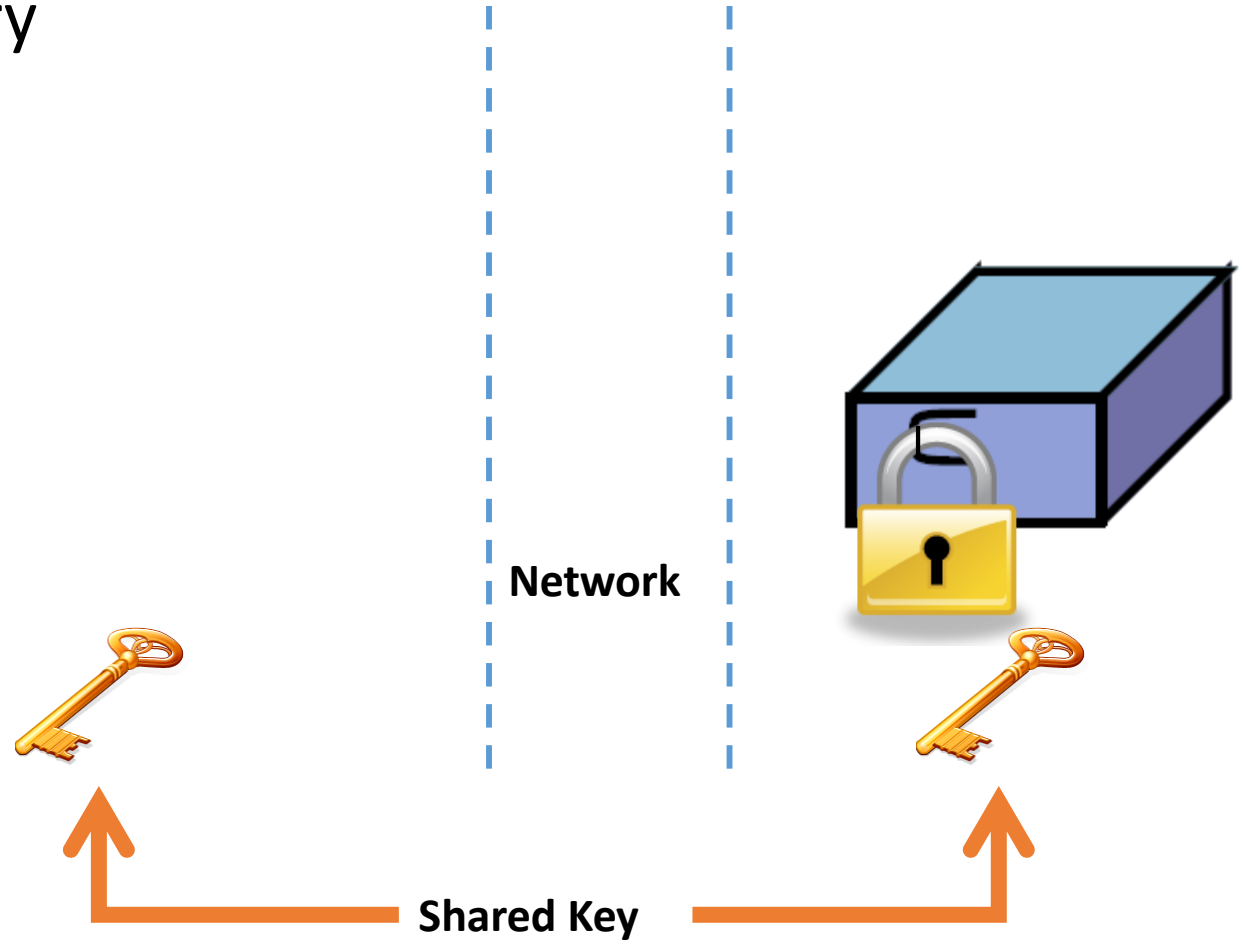
Symmetric Encryption

- Message delivery



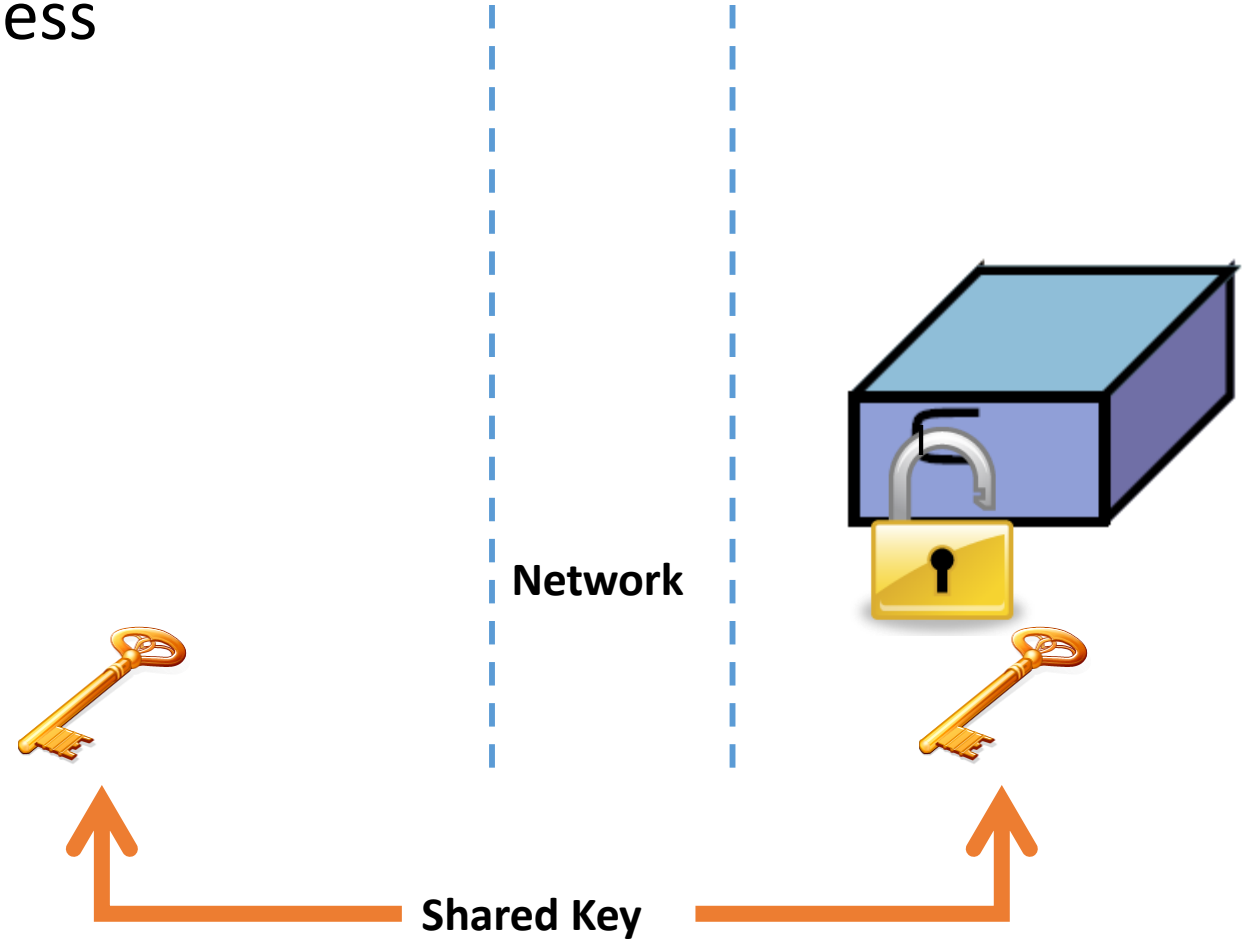
Symmetric Encryption

- Message delivery



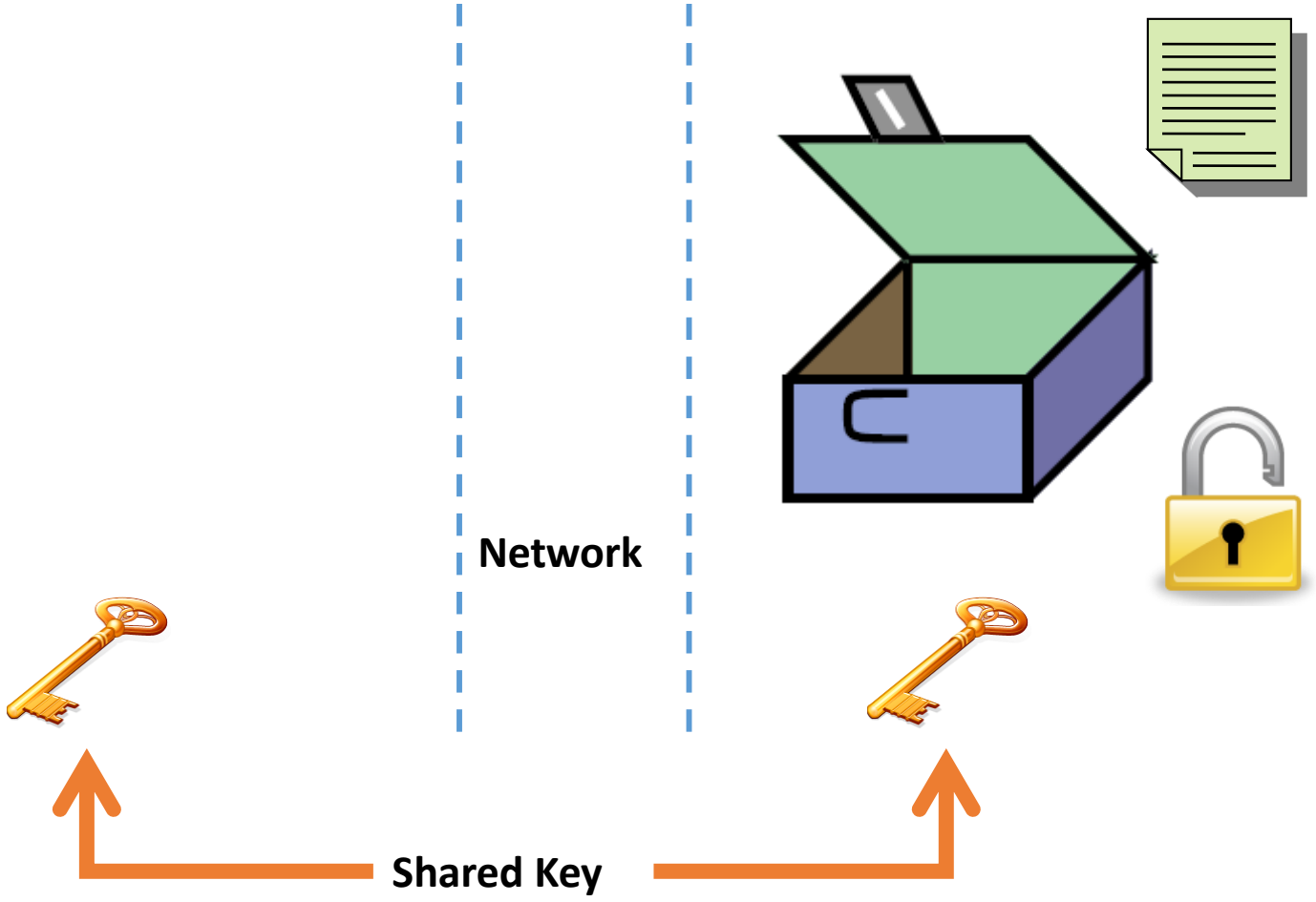
Symmetric Encryption

- Decryption process



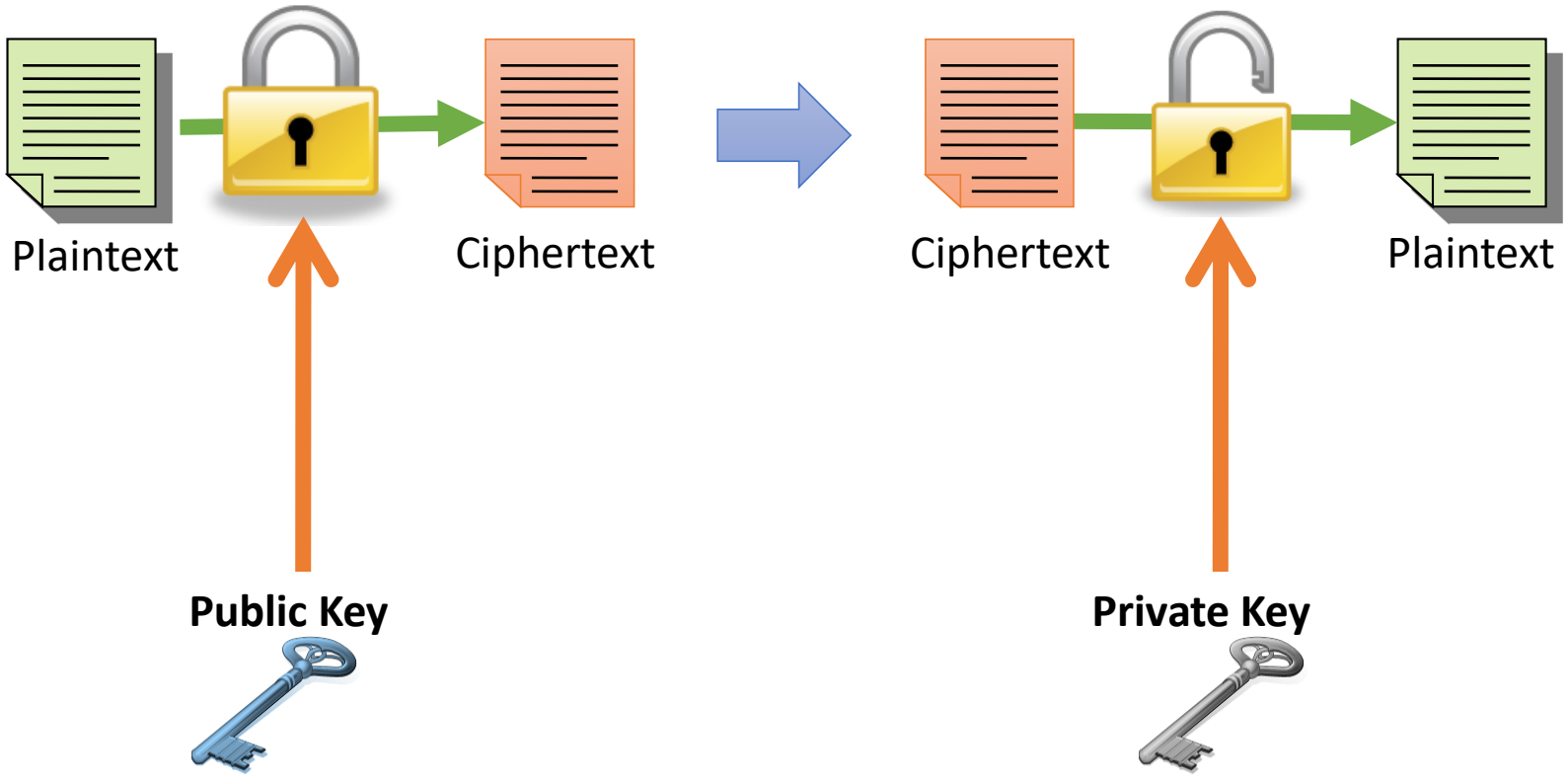
Symmetric Encryption

- Plaintext

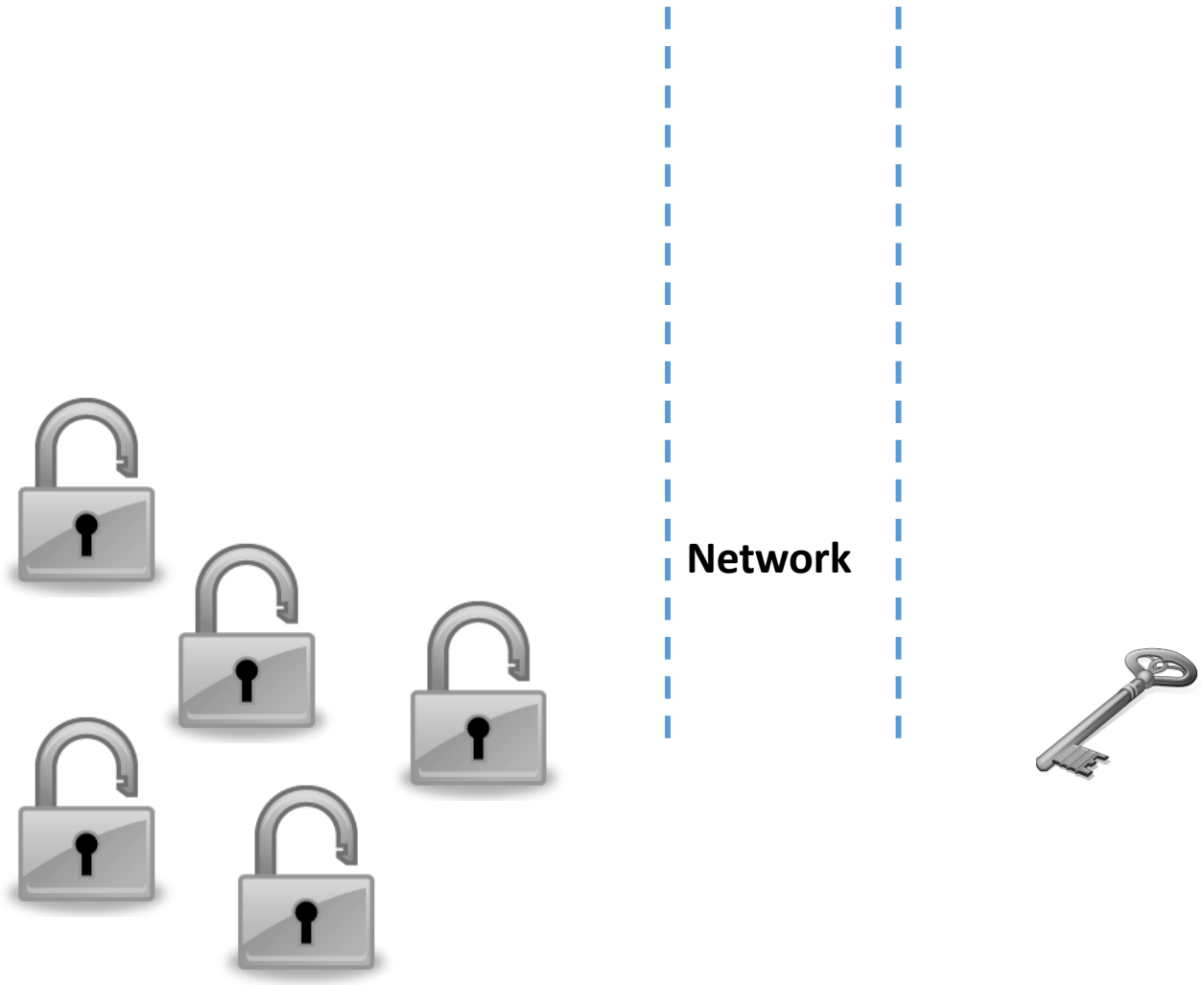


Public-Key Encryption

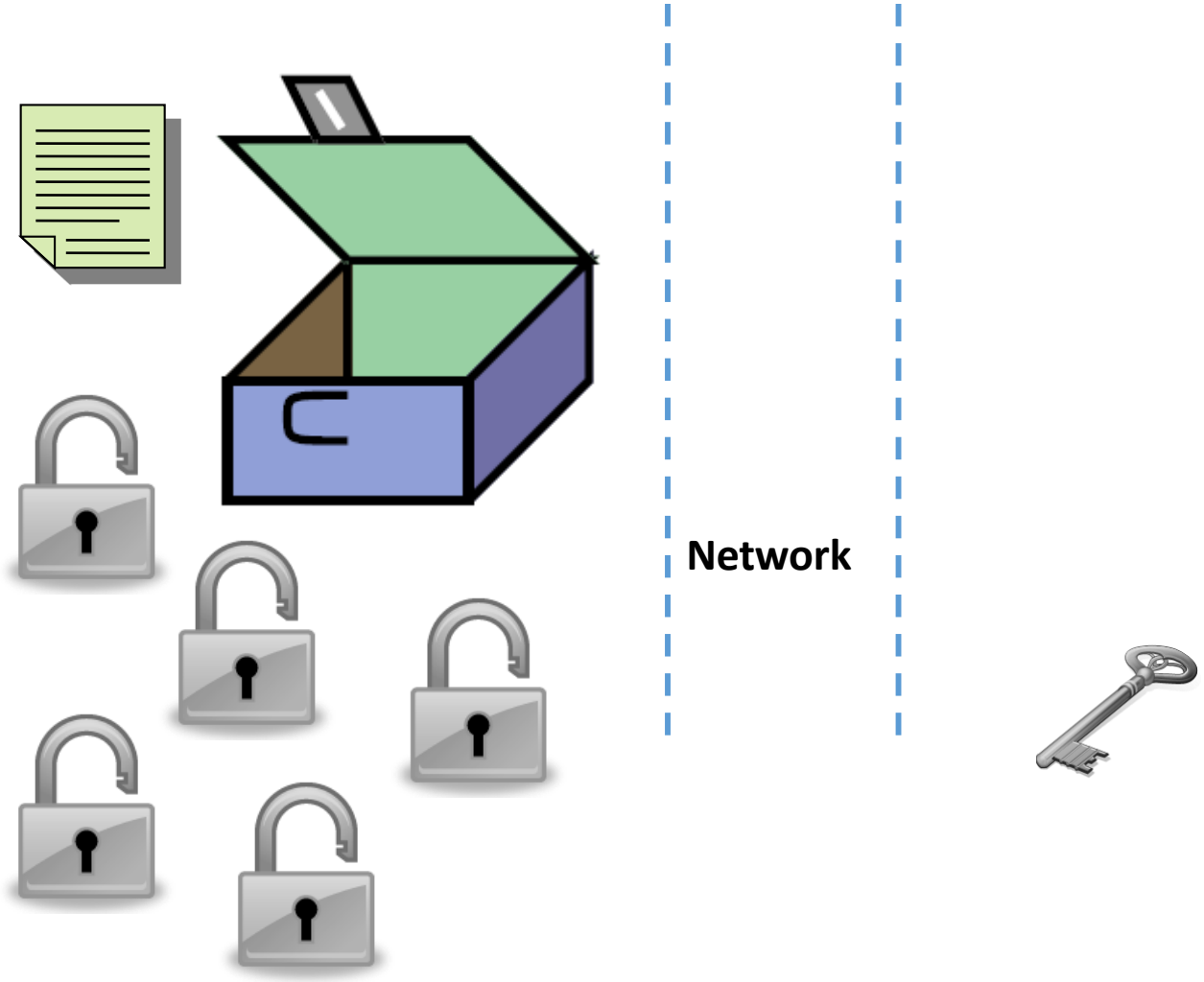
- Encryption and decryption use different keys



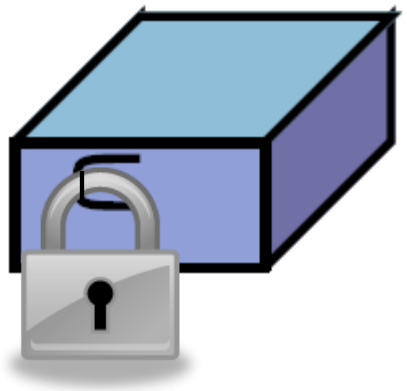
Public Key Encryption



Public Key Encryption

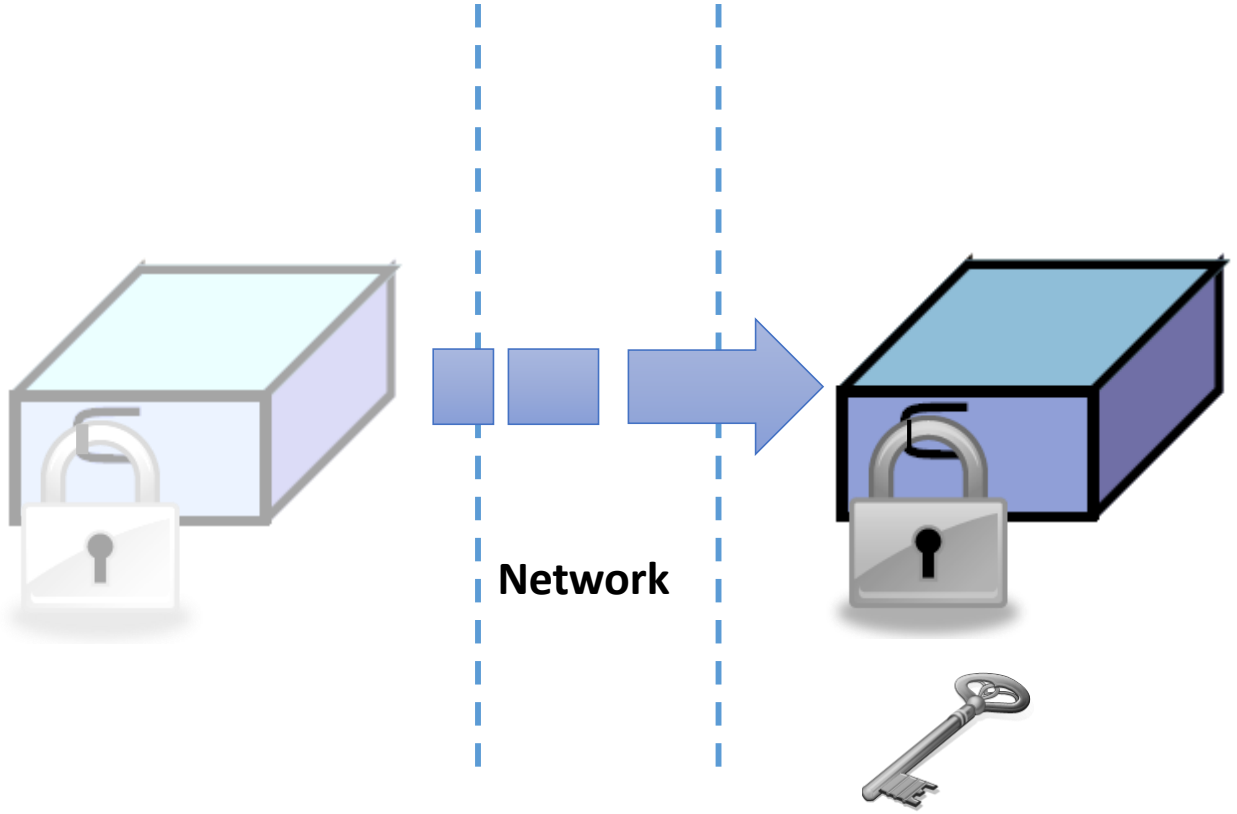


Public Key Encryption

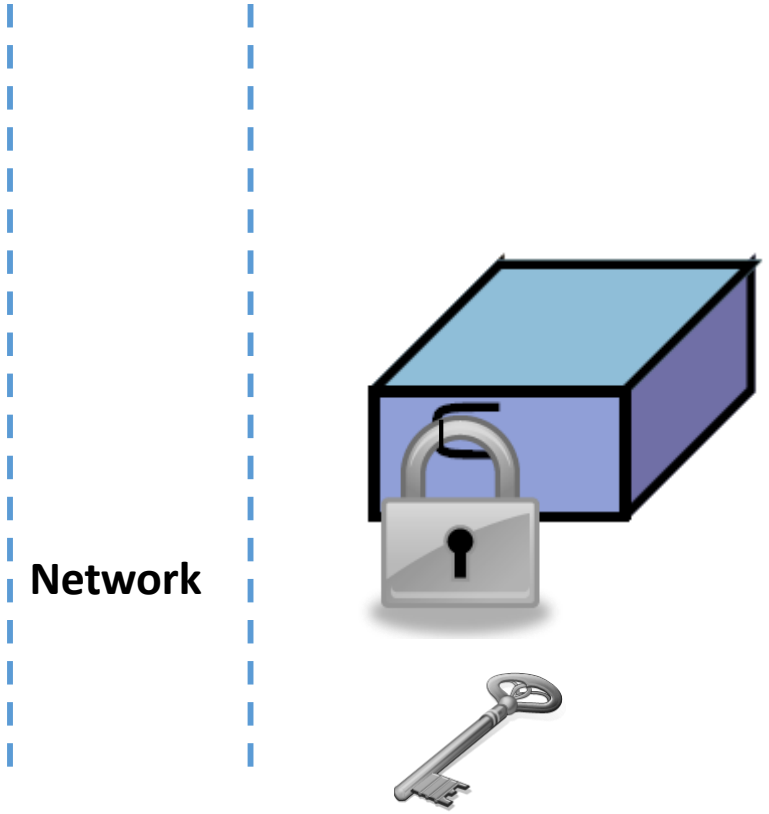


Network

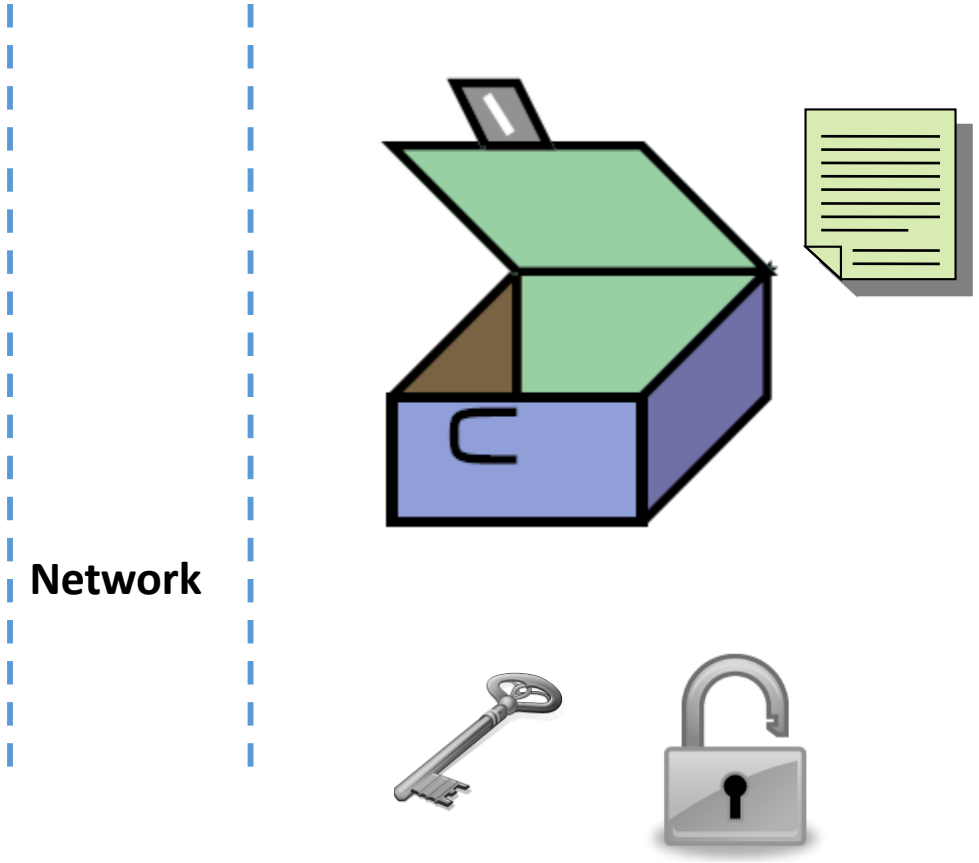
Public Key Encryption



Public Key Encryption



Public Key Encryption



Advantages and Disadvantages

- **Advantages**
  - No need to share the same key

- **Disadvantages**
  - How to trust who’s the lock’s holder?
  - Who originated the document?

Public-Key Encryption

- **Requirements:**
  - Easy to generate pairs of key.
  - If ***Alice*** knows the public key of ***Bob,*** it should be easy for Alice to generate the ciphertext ***c*** of any plaintext message ***m*** of her liking (intended for Bob).
  - It should be easy for ***Bob*** to decrypt the message intended for him using his private key.
  - It is computationally infeasible for an opponent, knowing the ciphertext and public key, to recover the original message ***m*** or the **private key**.

Public-Key Encryption

- **Operation principle**
  1. Each user generate two keys, public and private.
  2. The user put their public key in public register. These public keys are accessible by anyone. Users keep their private keys, private!
  3. If Alice wishes to send a message to Bob, she encrypts the message using Bob’s public key.
  4. Bob decrypts it using his own private key. No one else could decrypt the message because his private key is private.

Modular Arithmetic

Modular arithmetic is a system of arithmetic for integers, where numbers "wrap around" upon reaching a certain value;

A good example is the clock!

$a = n \times q + r$

where r is the reminder or $r = a \bmod q$ & $n \in I$

e.g. $14 \bmod 12 = 2 \rightarrow 14 = 1 \times 12 + 2$

$26 \bmod 12 = 2 \rightarrow 26 = 2 \times 12 + 2$

Since the hour number starts over after it reaches 12, this is arithmetic modulo 12.

12 is **congruent** not only to 12 itself, but also to 0, so the time called "12:00" could also be called "0:00", since $0 \equiv 12 \bmod 12$

So, what is the Congruent? ➔

The congruence relation

- A congruence relation (or simply congruence) is an equivalence relation on an algebraic structure.

- Basic Example:

  If a and b satisfy $a \bmod n = b \bmod n$, then $a \equiv b \pmod{n}$

  e.g. 73 mod 23 = 4 and 4 mod 23 = 4

  then $73 \equiv 4 \pmod{23}$

One-Way Functions

- A one-way mathematical function is very easy to do, but very difficult to reverse.
- *Examples*
    - **Easy to do:**

      7919 x 7927 = 62 773 913
    - **Difficult:**

      What are the factors of 1 689 259 081 189
- Public-key encryption can be thought of as a function that anyone should be able to compute, since the encryption key is public.
- And it should be very difficult for an attacker to efficiently determine a plaintext from knowledge of a ciphertext and the public encryption key.

Only 1299709 x 1299721

Examples of the difficulty of factoring the product of two primes

Challenge number	Difficulty of factoring to the two primes
15	Everyone can do this instantly
143	Doable with a little thought
6887	Should not take more than a few minutes
31897	A calculator is now useful
20-digit number	A computer is now required
600-digit number	This is impossible in practice
600-digit even number	One factor immediate, other easily computed
600-digit number with small factor	One factor easily found, other easily computed

Trapdoor One-Way Function

- We know a one-way function is easy to compute, but whose inverse is intractable
- We want more for public-key encryption, though
- we need to be able to compute the inverse of the one-way function if we have the private key
- So a **trapdoor one-way function** is a one-way function *f* , together with a secret *y*, such that, given *f(x)* and *y*, it is easy to compute *x*
- *The private key here is our trapdoor.*

Modular Exponentiation with a Large Modulus

- Modular exponentiation thus involves raising a number *a* to the power *b* and then calculating the result modulo *n*, for some number *n*.

$$f(b) = a^b \bmod n$$

- Given numbers *a* and *n* (where *n* is prime), the function $f(b) = a^b \bmod n$, is believed to be a one-way function, since computing $f(b)$ is easy.
- But, given $f(b)$, working out what *b* is appears to be hard, assuming *n* is large. This difficult problem is often referred to as the *discrete logarithm problem*.

RSA Public-Key Algorithm

- The RSA algorithm was published by Rivest, Shamir and Adleman in 1977
- Invented independently at GCHQ earlier in 1973, but was not revealed until 1997 due to its top-secret classification.
- Based on the practical difficulty of factoring a very large number, i.e. solving the following problem: Given a large n = pq with p and q being prime, find p & q
- Can be used both for encryption and digital signature.

RSA algorithm

- RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and ***n***-1.

- So the first step is to represent the plaintext block in terms of an integer between **0** and **n-1**.

RSA algorithm: Encryption and Decryption

- Encryption
  - $c = m^e \bmod n$
  - (e, n) **: public key**
- Decryption
  - $m = c^d \bmod n$
  - d **: private key**
  - sometime, the pair (n,d) is called the RSA's private key, but the "private" part is really "d" (why?)

RSA algorithm: Correctness

- But does the decryption really recover m?
- That is: how do we get:

  $((\boldsymbol{m^e}\ \textbf{mod}\ \boldsymbol{n})^{\boldsymbol{d}}\ \textbf{mod}\ \boldsymbol{n}) = \mathrm{m}$

  For any m (0 <= m <= n-1)?

RSA algorithm: Correctness

- Turns out (thanks to number theory), that if the three integers (e, n, d) are <u>chosen carefully</u>, we will indeed have:

  $((\boldsymbol{m^e}$ **mod** $\boldsymbol{n})^{\boldsymbol{d}}$ **mod** $\boldsymbol{n})$ = m

  For any m (0 <= m <= n-1)!

RSA algorithm: Correctness

- The math behind it is actually not new per se: It had been known since the time of Fermat and Euler!

RSA

- What does this special relation between n, e, and d look like?
- we should have:
  - n = p x q (p and q are large primes).
  - For e, choose any number e that is relatively prime to z = (p - 1) x (q - 1). (e has no common factors with z).
  - To find d, solve the equation: $ed \equiv 1$ (mod z). That is, ed is an element divisible by e in the series z + 1, 2z + 1, 3z + 1, ... kz + 1... (done using an algorithm known as the *Extended Euclidean Algorithm*)
- The pair is generated starting from p and q then finding e, d. z should be discarded, n and e are kept public, and d is kept private.

RSA example

p = 13, q = 17

$n = p$ x q = 221

$z = (p - 1)(q - 1)$ = 192

choose e = 5

$ed \equiv 1 \bmod z$, $ed \equiv 1 \bmod 192$, so ed can be 1; 193; 385; ...

385 is divisible by 5

d = 385/5 = 77

Encryption is $m^5 \bmod 221$ and Decryption is $c^{77} \bmod 221$.

- If p = 3, q = 11, e = 7, plaintext m = 5, calculate d and ciphertext c.

Why RSA Works

- Suppose (n,e) and (n,d) are key pair and $m<n$.

Then $m = c^{\mathrm{d}} \bmod(n) = (m^{\mathrm{e}})^{\mathrm{d}} \bmod n = m^{\mathbf{ed}} \bmod n$

- To prove $(m^{\mathrm{e}})^{\mathrm{d}} \equiv m \bmod n$

$$
\begin{aligned}
m^{\mathrm{ed}} &= m^{(1+k(p-1)(q-1))} \\
&= m(m^{(p-1)(q-1)})^{k} \\
&\equiv m(1^{k}) \bmod n \\
&\equiv m \bmod n
\end{aligned}
$$

But $0<m<n$, so congruence mod n implies genuine equality.

Security of RSA

Decrypting a Ciphertext Without Knowledge of the Private Key

$$c = m^e \bmod n$$

Computing m from c, e, and n is believed (but never proved) to be a hard problem, called “the RSA problem”, and thus the encryption function of RSA is considered a one-way function (for now!).

Security of RSA

Determining the Private Key Directly from the Public Key

To determine a private key d from a public key (n, e), an attacker need to:

1. Factor $n=pq$, into its prime factors p and q are; and
2. Run the Extended Euclidean Algorithm to determine d from p, q, and e.

But, remember: factoring the product of two large prime numbers is believed (again, not proved!) to be a hard problem!

Security of RSA

Deterministic Cipher (same problem with ECB):

Introducing randomness: RSA-OAEP

ElGamal and Elliptic Curve (EC) Cryptosystem

- Turns out, RSA is not the only way to achieve a trapdoor one-way function (and hence public key cipher).
- In fact, there is a whole family of them called “ElGamal”, and its “Elliptic Curve” (EC) variants. Some of them even provide some benefits over RSA.
- We will only discuss ElGamal, because a similar idea is going to show up later (Diffie-Hellman key exchange), and leave EC to the interested (not part of evaluation!)

ElGamal: in 1985 by Taher ElGamal

ElGamal: Encryption

We need four parameters: p, g, y, x

ElGamal Encryption of message m:

1.Randomly generate a number k;
2.Compute C_1 and C_2 as:

- $C_1 = g^k \bmod p$
- $C_2 = m \times y^k \bmod p$; then
- ciphertext $C=(C_1, C_2)$

- public key: (p, g, y)

ElGamal: Decryption

ElGamal Decryption of ciphertext C = (C_1, C_2):

1. Compute $C_1^x \bmod p$; and
2. Divide C_2 by the result of step 1:

$$\mathbf{m} = C_2 / C_1^x \bmod p.$$

3.Private key: x

ElGamal: Correctness

As with the RSA, the first question is, what special relations between public key (p, g, y) and private key x there should be, so that the decryption of the ciphertext gives back the plaintext (these detail are not part of evaluation):

- p: a large prime (in the order of 3072 bits);
- g: a *primitive element modulo p.* For the sake of this module, just think of it as "specially tangled with p"!
- x: a randomly selected number between 1 and p-1;
- y: also tangled with p, g, x as follows:

$$y = g^x \bmod p$$

ElGamal: Correctness

Then one can show you can get back the message from decryption (so ElGamal works!). The proof is in the appendix and will not be part of evaluation:

$$C_1^{\,x} = (g^k)^x = (g^x)^k = y^k \bmod p$$

$$C_2 \times (y^k)^{-1} = (m \times y^k) \times (y^k)^{-1}$$
$$= m \times (y^k \times (y^k)^{-1}) \bmod p = m \times 1 = m \bmod p$$

ElGamal: Security

Difficulty of Decrypting a Ciphertext Without Knowing the Private Key:

The adversary has the public key (p, g, y) and sees the ciphertext $C_1 = g^k \bmod p$, and $C_2 = m \times y^k \bmod p$,

So, can the attacker derive k from $(g^k \bmod p)$, knowing g and p?

ElGamal: Security

Difficulty of Determining the Private Key Directly from the Public Key:

Again, the adversary has the public key (p, g, y) and knows the "special relation" of $y = g^x \bmod p$.

So, can the attacker derive x from $(g^x \bmod p)$, knowing g and p?

ElGamal vs RSA: Security Summary

Security of RSA:

- Difficulty of decryption without private key == Difficulty of taking the discrete e'th root, i.e., solving $\boldsymbol{m^e}$ **mod** $\boldsymbol{n}$ $for\ m$.
- Difficulty of deriving private key from public key == Difficulty of prime factorization of a large integer.

Security of ElGamal:

- Difficulty of decryption without private key == Difficulty of discrete logarithm, i.e., solving $\mathbf{g^k}$ **mod p** $for\ k$.
- Difficulty of deriving private key from public key == Difficulty of discrete logarithm, i.e., solving $\mathbf{g^x}$ **mod p** $for\ x$.

Problem with Public-key Encryption

- **Computational costs**: slower than DES by a factor of a thousand!
- **Long-plaintext security issues**
  - For longer plaintext → split into “blocks” and then encrypt these separately.
  - Essentially encrypting these blocks using the public-key equivalent of ECB mode for a block cipher, which is not desirable.
  - However, there are no alternative modes of operation proposed for public-key encryption.

Problem with Public-key Encryption

- Symmetric cryptography and Public-key cryptography have both strength and weakness.
- In particular, we often need the good properties of both:
  - Ease of key sharing from Public-key cryptography.
  - Efficiency and simplicity from Symmetric cryptography.

- *So what should we do?*

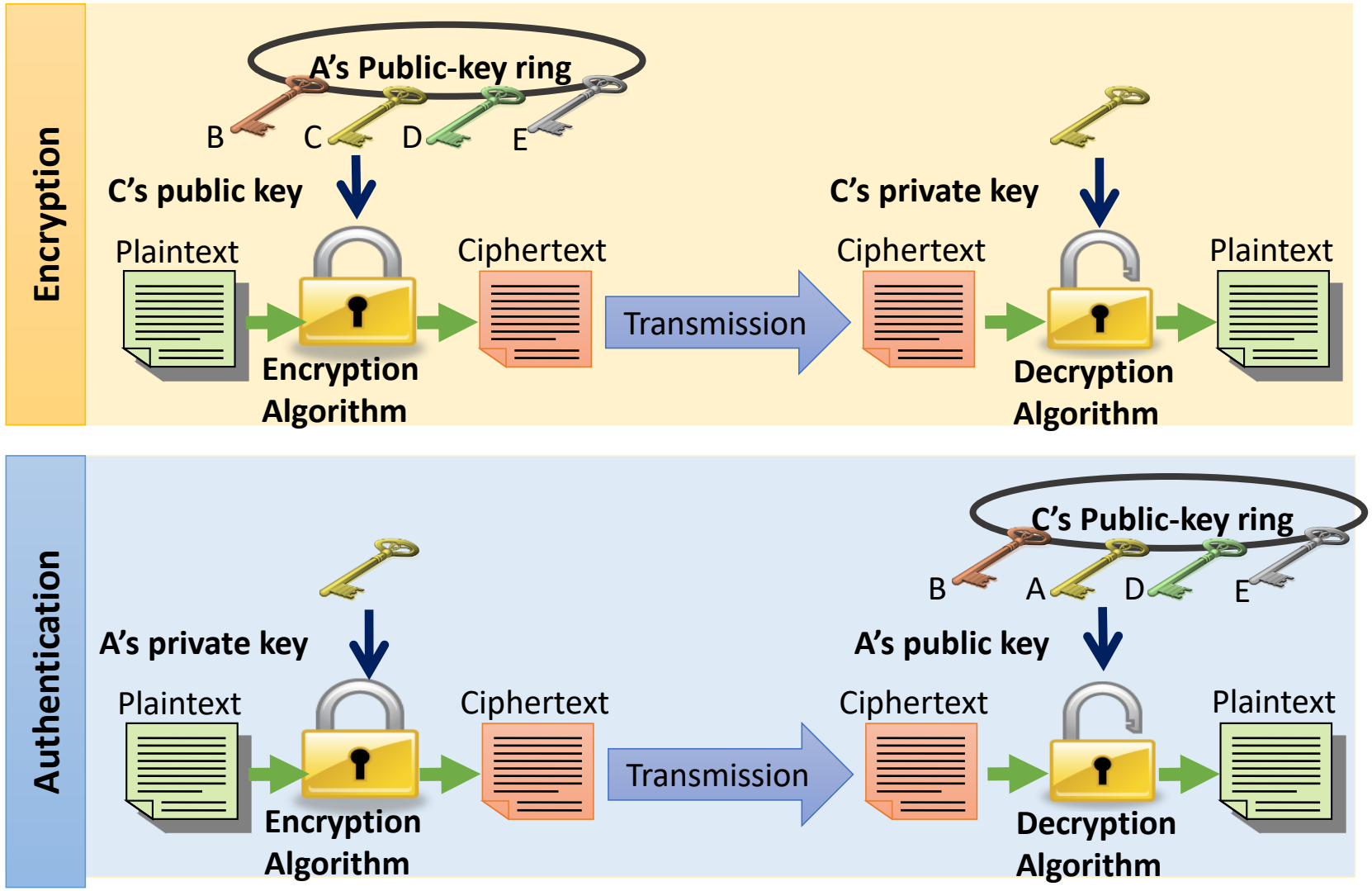
Hybrid Encryption

For secure communications with potentially high traffic, the **hybrid cryptography** gives an ideal answer.

Hybrid Encryption

- Alice wishes to exchange several messages with Bob securely. We assume Alice and Bob share their public keys.
- First, Alice creates a fresh session key (symmetric encryption key), K_{AB}.
  1. A -> B : $\{K_{AB}\}K_{B;\ pub}$
  2. B -> A : $\{M1\}K_{AB}$
  3. A -> B : $\{M2\}K_{AB}$
  4. B -> A : $\{M3\}K_{AB}$
- (In Step 2, we assume Bob decrypts the message and obtains K_{AB}.)

Authentication with Public-key Encryption



Summary

- Public-key Encryption
- One-way function
- Trapdoor one-way function
- RSA
- ElGamal
- Hybrid Encryption