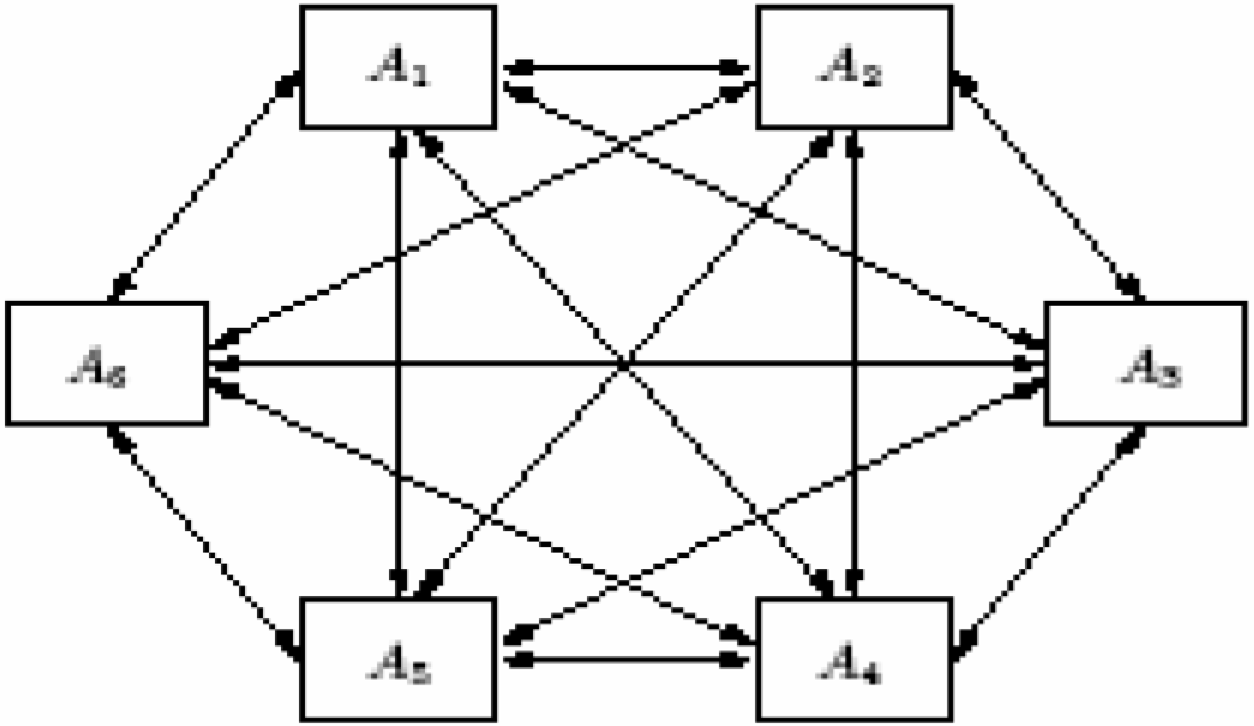
Public Key Cryptography (PKC)

Introduction

- Traditional private/secret/single key cryptography uses one key
  - shared by both sender and receiver - **symmetric, parties are equal**
  - does not protect the sender,
    - receiver can forge a message & claim that it has sent by sender
- if this key is disclosed, communications are compromised
- Therefore, a secure channel is required
  - to secretly transfer the key to receiver
- How to establish the secure channel – a practical problem
- Why can’t the message itself be communicated through this ?

PKC - Motivation

- How many pairs of keys are required for say *n users ? (symmetric key)*



PKC - Motivation

- total of $(n^2 - n)/2$ potential pairs: who wish to communicate privately !!
- it is unrealistic to assume that $(n^2 - n)/2$ pairs can be arranged
- PKC was proposed as
  - communication over a public channel
  - using publicly known techniques

PKC

- PKC is modern cryptography
  - probably most significant advance in the 3000 year history of cryptography
  - uses two keys – a public & a private key
  - asymmetric since parties are not equal
  - uses clever application of number theoretic concepts to function

- developed to address two key issues:
  - key distribution – how to have secure communications in general without having to trust a KDC with your key
  - digital signatures – how to verify a message comes intact from the claimed sender

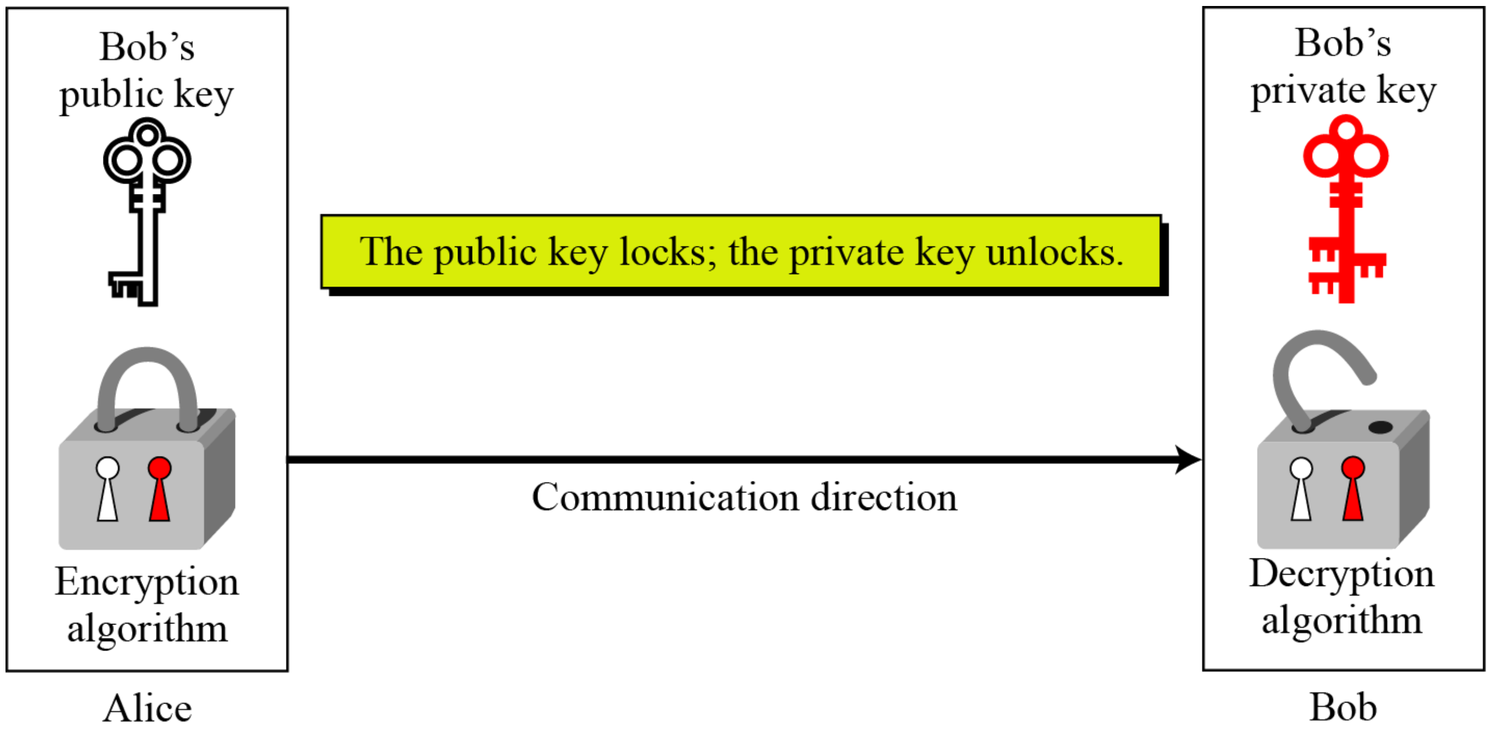
PKC

- Symmetric and asymmetric-key cryptography will exist in parallel and continue to serve the community.
  - they are complements of each other;
  - the advantages of one can compensate for the disadvantages of the other.

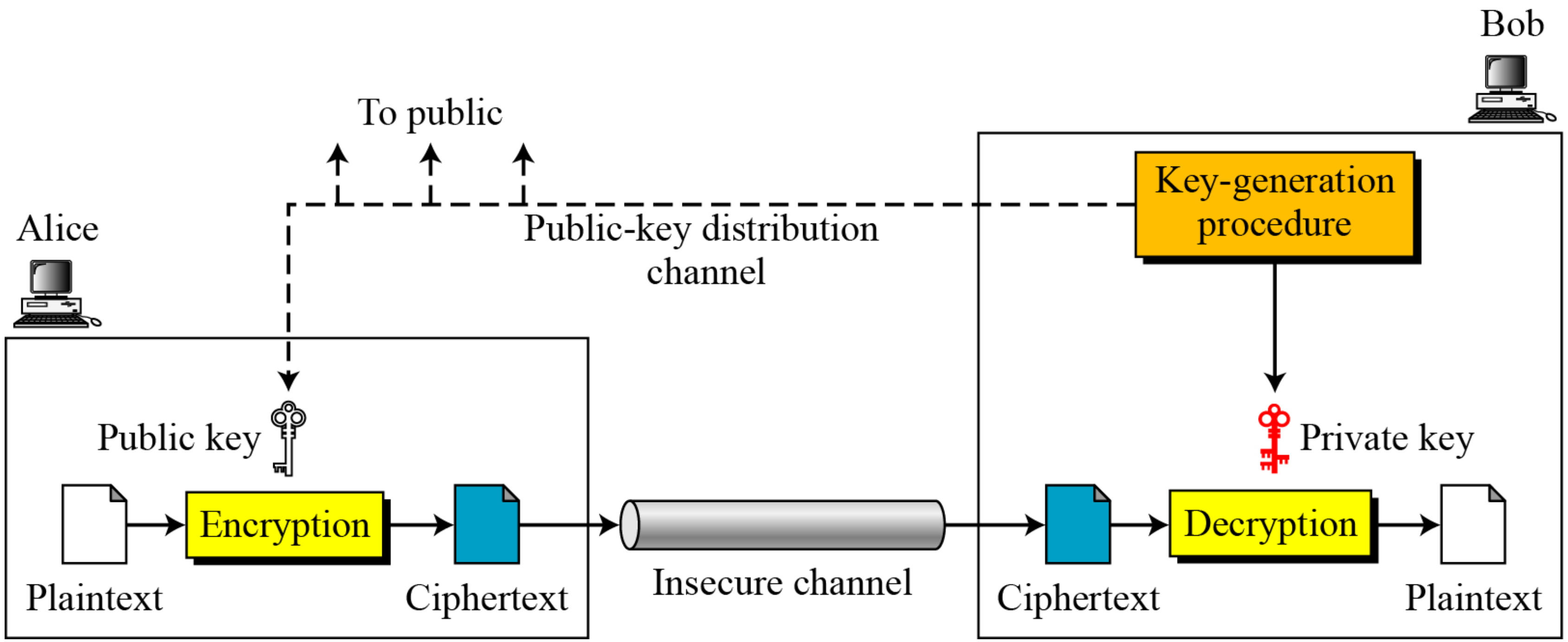
Symmetric-key cryptography is based on sharing secrecy;
Asymmetric-key cryptography is based on personal secrecy.

PKC

- Asymmetric key cryptography uses two separate keys: one private and one public.

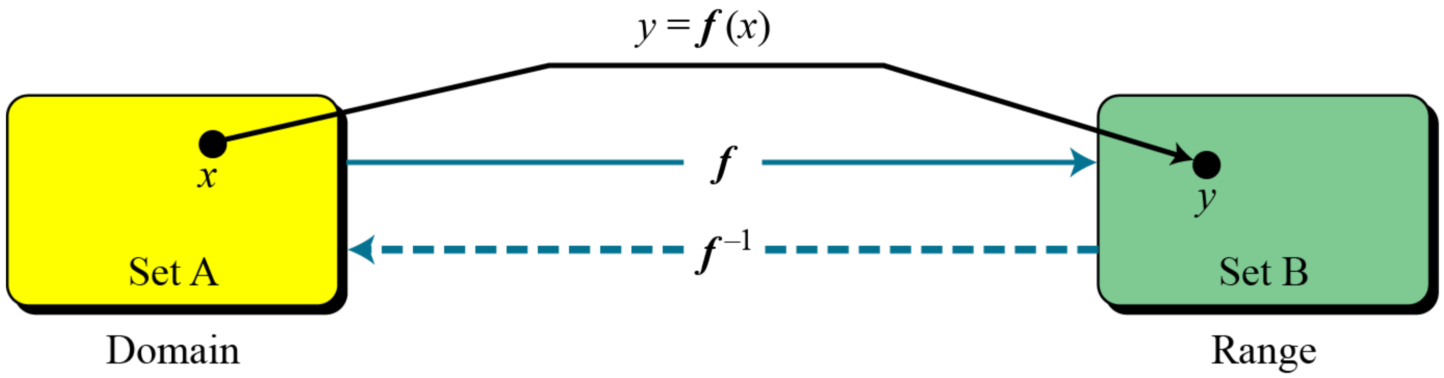


General Idea of PKC



PKC

- Plaintext/Ciphertext
  - Unlike in symmetric-key cryptography, plaintext and ciphertext are treated as integers in asymmetric-key cryptography.
- The main idea behind asymmetric-key cryptography is the concept of the trapdoor one-way function.



A function as rule mapping a domain to a range

PKC

- One-Way Function (OWF)

1. f is easy to compute.
2. f^{-1} is difficult to compute.

- Trapdoor One-Way Function (TOWF)

3. Given y and a trapdoor, x can be computed easily.

Example

- **Example 1:**
  - When n is large, $n = p \times q$ is a one-way function.
  - Given p and q , it is always easy to calculate n ;
  - given n, it is very difficult to compute p and q. This is the factorization problem.

- **Example 2:**
  - When n is large, the function $y = x^k \bmod n$ is a trapdoor one-way function.
  - Given x, k, and n, it is easy to calculate y.
  - Given y, k, and n, it is very difficult to calculate x.

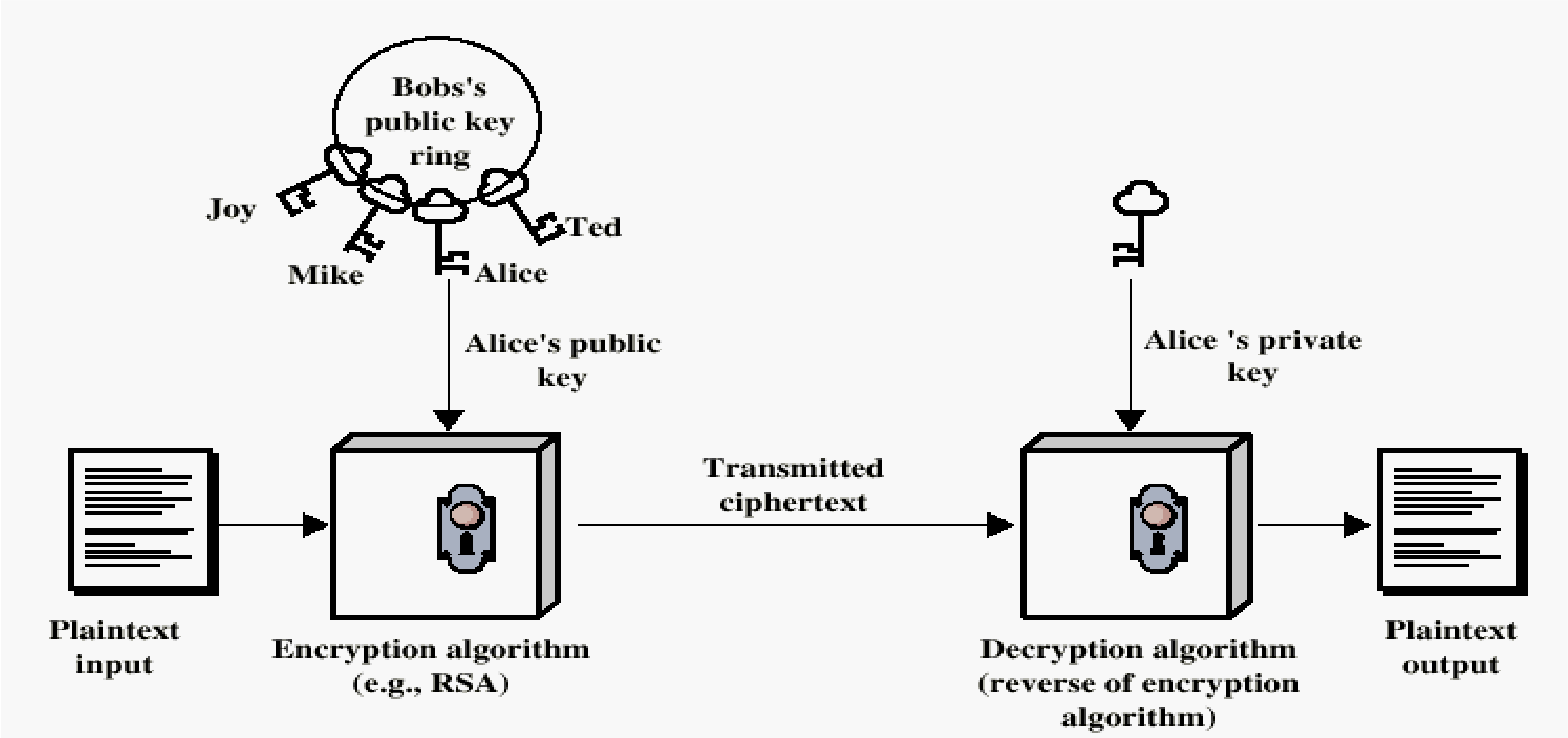
Example

- 1024 bit Prime Number:

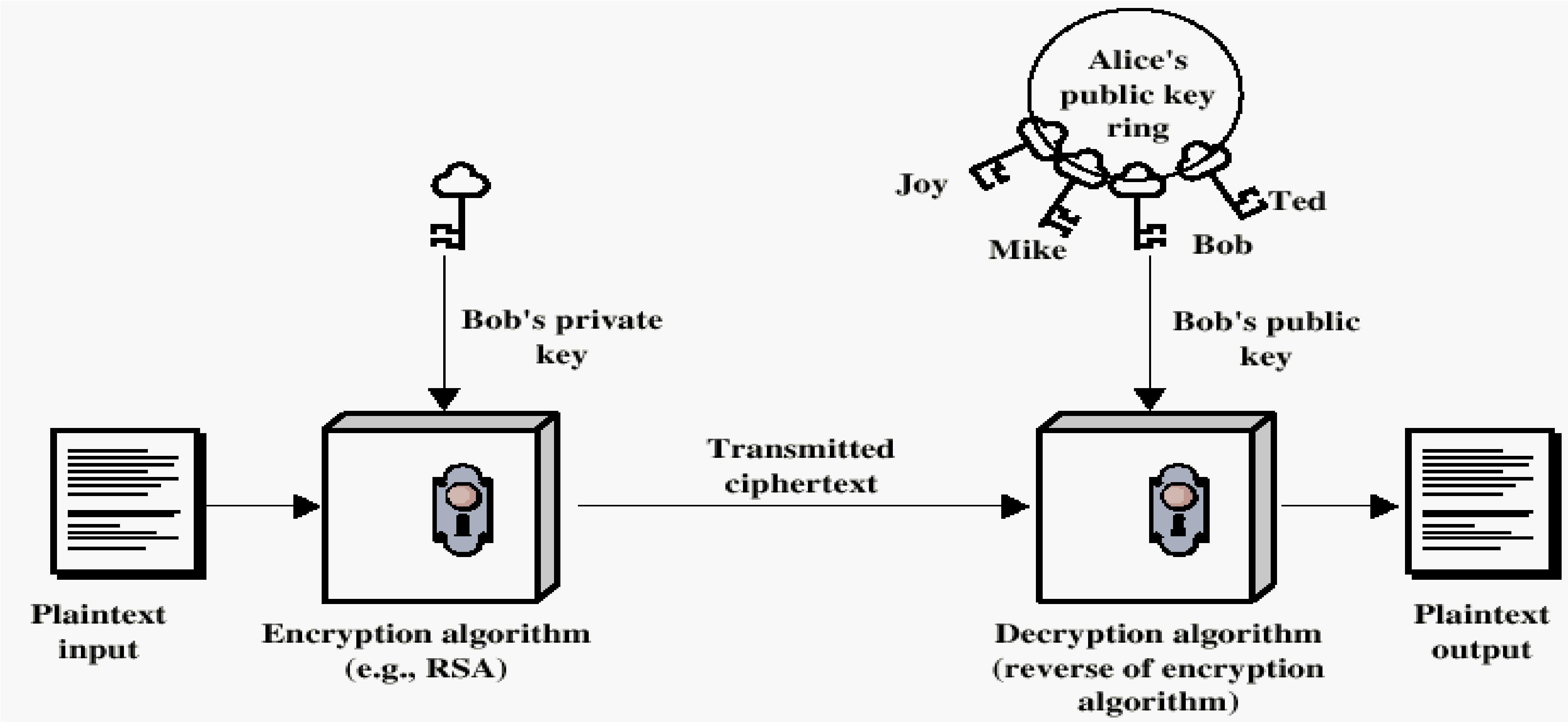
14926660406676521425746589984505259593698043308528112047243863356010910984506208081319567489713652594984018496531250529886994872297764946902308436155041298948606020791758
05404540811405873538622344455775204768725436764861678924438723087050267784611212612243224953283466303834863866636288787728384490877770123303

- Now, In real life applications, 2048 bit prime numbers are used.

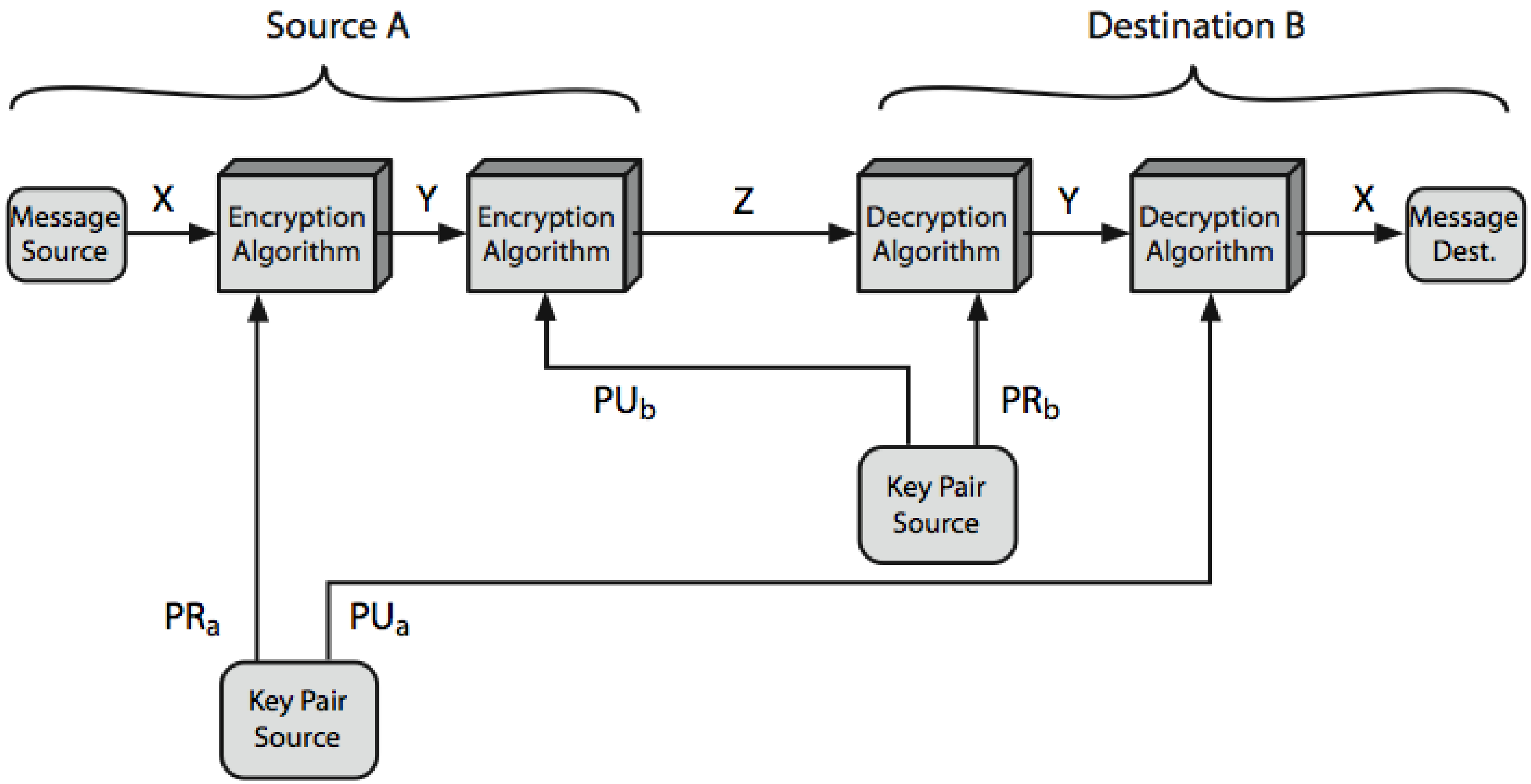
Asymmetric Encryption



PKC Authentication



PKC – Encryption & Authentication



PKC Applications

- can classify uses into 3 categories
  - encryption/decryption
    - the sender encrypts a message with the recipient's public key.
  - digital signature
    - the sender "signs" a message with its private key.
  - key exchange
    - two sides cooperate two exchange a session key.

- some algorithms are suitable for all uses, others are specific to one

Public key Characteristics

- Public-Key algorithms rely on two keys where:
  - it is computationally infeasible to find decryption key knowing only algorithm & encryption key
  - it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)

- a problem being computationally easy means
  - it can be solved in polynomial time as a function of its input n i.e.
    - if the length of the input is **n** bits,
      - then the time to compute is proportional to n^a (a = some constant value)

Public key Characteristics

- computationally infeasible is difficult to define
- a problem is infeasible to solve
  - if grows faster than the polynomial time as a function of input size
    - i.e., if the length of the input is $\mathbf{n}$ bits, then
      - the time to compute is proportional to $\mathbf{2^n}$
- A one way trap door function

Security of public key

- brute force exhaustive search attack is always possible
  - like private key schemes
  - but keys proposed and used are too large (>1024bits)
    - For example: p=170141183460469231731687303715884105727,
  - renders brute force attack impractical
  - solution (security) relies
    - on a large enough difference in difficulty
    - between easy (en/decrypt) and hard (cryptanalyse) problems