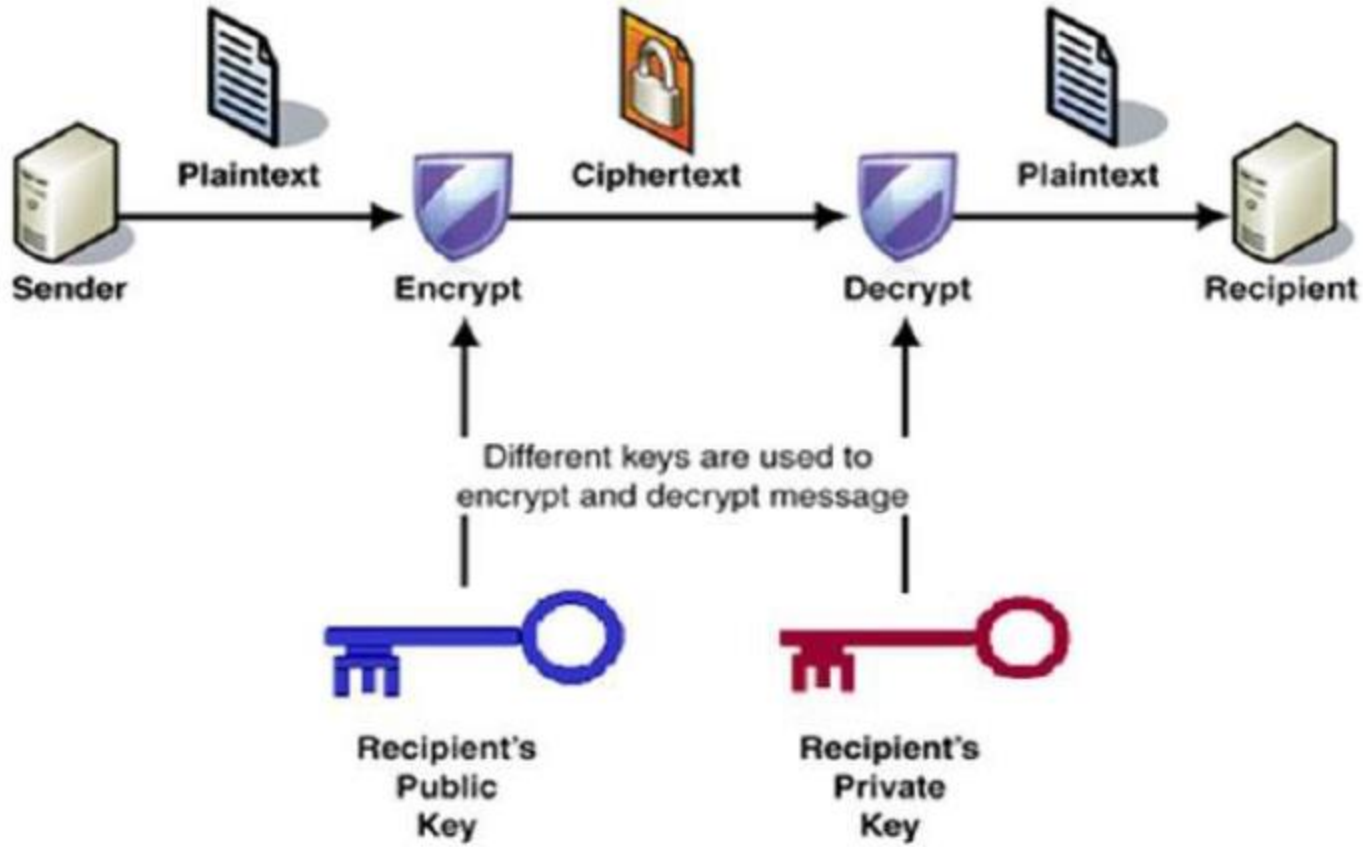
PUBLIC KEY ENCRYPTION



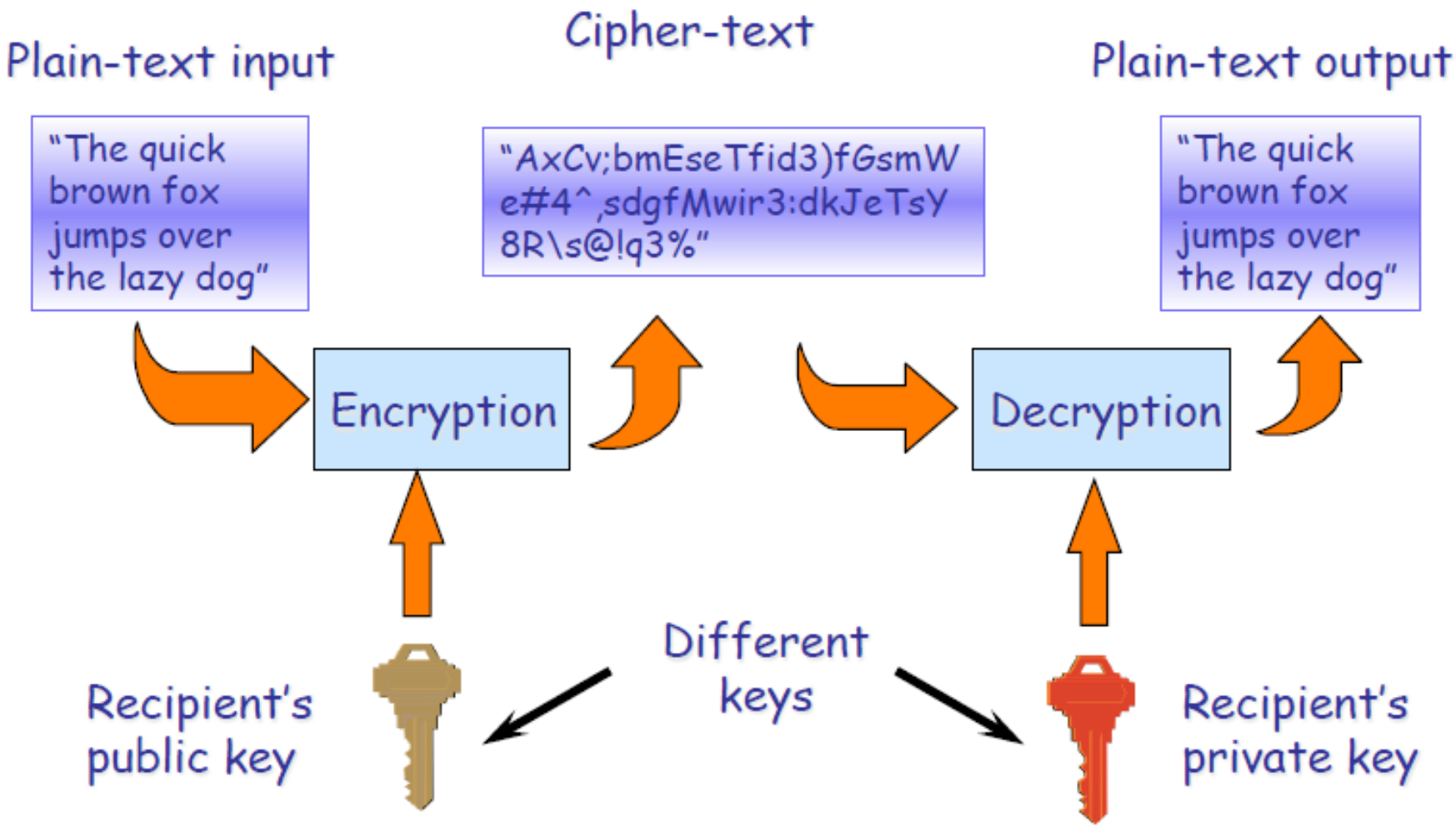
The most important properties of public key encryption scheme are −

- Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
- Each receiver possesses a unique decryption key, generally referred to as his private key.
- Receiver needs to publish an encryption key, referred to as his public key.
- Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
- Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the ciphertext and the encryption public key.
- Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

Public-Key Cryptography

- **Public-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, distributed by the owner to anybody,
  - a **private-key**, known only to the owner.
- Each user will thus have a collection of public keys of all the other users.
- It is **asymmetric** because
  - keys used to encrypt messages **cannot** be used to decrypt them

Asymmetric Cryptography



Public-Key Cryptography

- The public key does not need not be secret
  - authenticity is required to guarantee that the key's owner is the only party who knows the corresponding private key
- A primary advantage of such systems is that providing authentic public keys is generally easier than distributing secret keys securely, as required in symmetric key systems.

Why Public-Key Crypto?

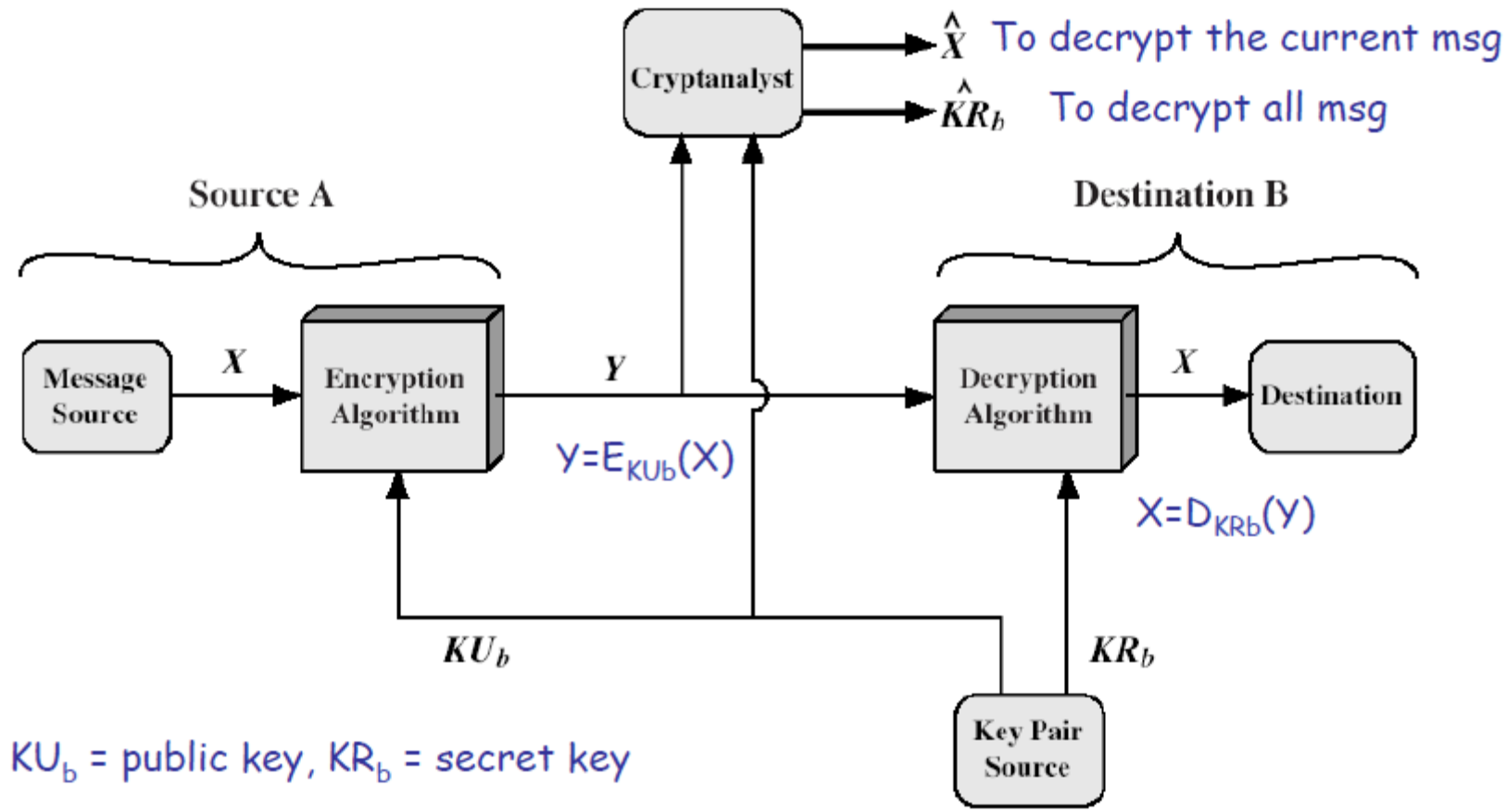
- It was developed to address two key issues:
  - **key distribution**
    - how to communicate securely without trusting a KDC
  - **digital signatures**
    - verify that a message is intact and comes from the claimed sender
- *Public* invention due to Diffie & Hellman at Stanford University in 1976
  - The concept had been previously described in a classified report in 1970 by James Ellis (UK CESG) - and subsequently declassified in 1987

Public-Key Applications

We can classify its uses into 3 categories:

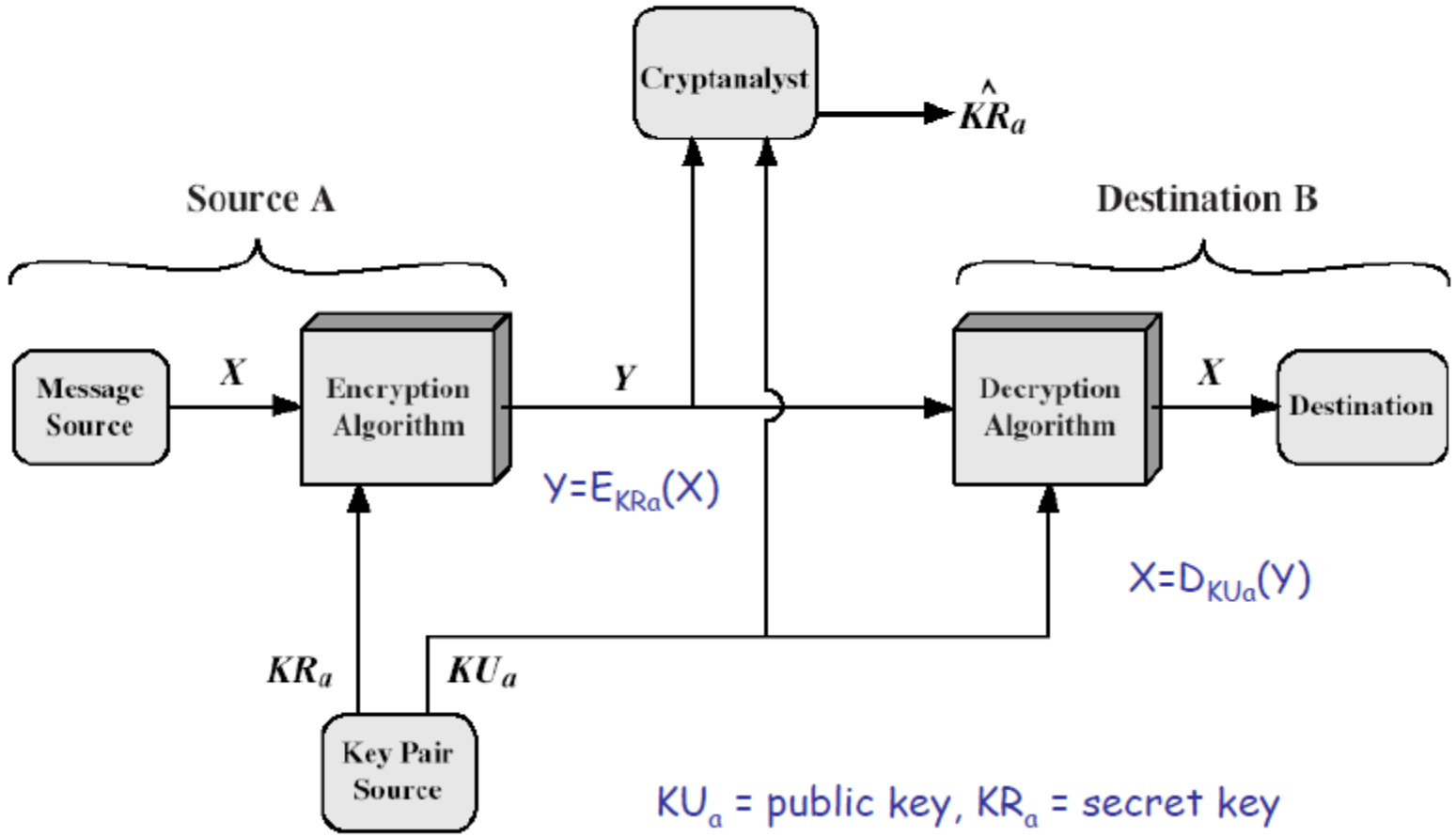
- **encryption/decryption** (secrecy)
  - sender encrypts the msg with recipient's public key
- **digital signatures** (authentication & data integrity)
  - sender encrypts msg with his/her private key
- **key exchange** (of session keys)
  - several approaches, using one or two private keys .

Confidentiality, key distribution

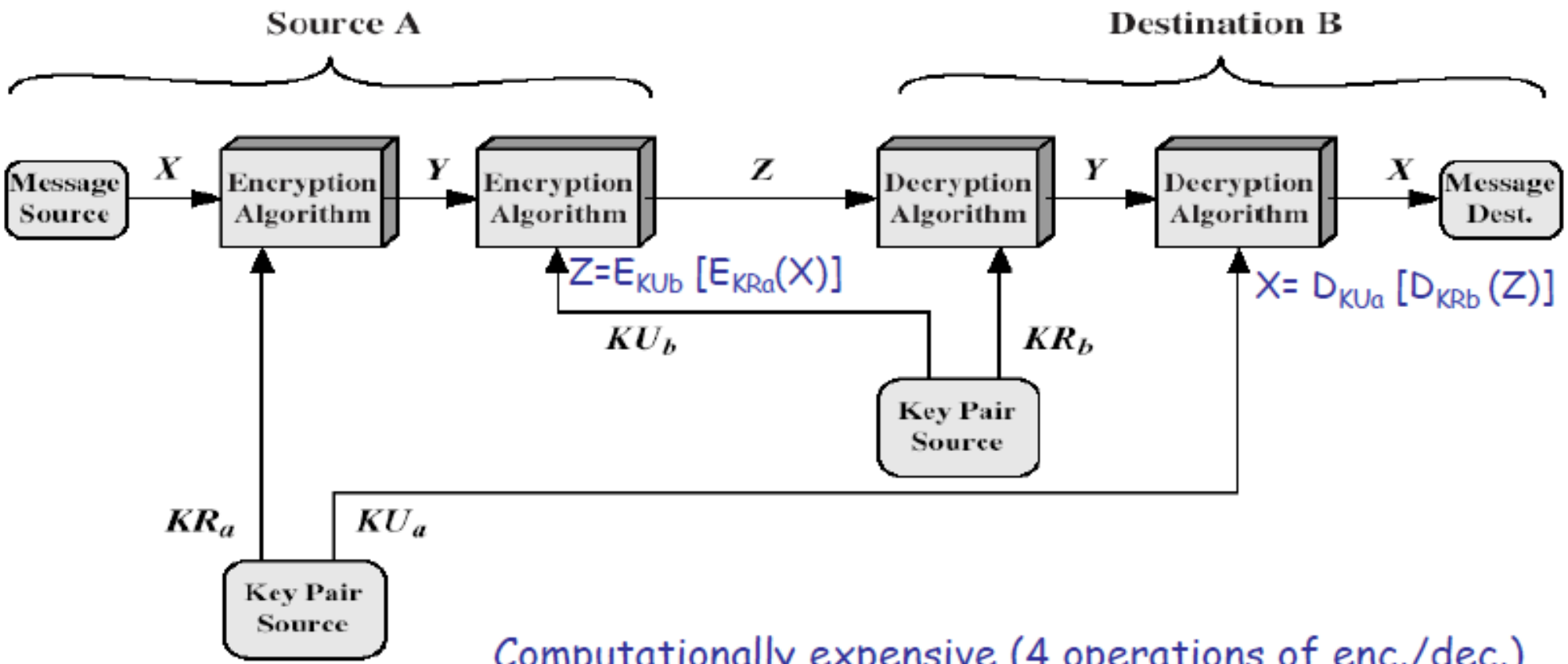


KU_b = public key, KR_b = secret key

Authentication, without confidentiality



Confidentiality and authentication



Computationally expensive (4 operations of enc./dec.)

RSA algorithm

- Invented by Rivest, Shamir & Adleman at MIT in 1977
- Best known & widely used public-key scheme
- Security based on the intractability of the integer factorization problem.

RSA algorithm

- Currently used in a wide variety of products, platforms, and industries around the world.
  - RSA is built into current operating systems by Microsoft, Apple, Sun, and Novell.
  - In hardware, RSA can be found in secure telephones, on Ethernet network cards, and on smart cards.
  - RSA is incorporated into all of the major protocols for secure Internet communications, including S/MIME, SSL, and S/WAN.

Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below −

- **Generate the RSA modulus** n
  - Select two large primes, p and q.
  - Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.
- **Find Derived Number** e
  - Number **e** must be greater than 1 and less than $p - 1q - 1$.
  - There must be no common factor for e and $p - 1q - 1$ except for 1. In other words two numbers e and $p - 1q - 1$ are coprime.

- **Form the public key**
    - The pair of numbers n, e form the RSA public key and is made public.
    - Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes p & q used to obtain n. This is strength of RSA.
- **Generate the private key**
    - Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
    - Number d is the inverse of e modulo p - 1q – 1. This means that d is the number less than p - 1q - 1 such that when multiplied by e, it is equal to 1 modulo p - 1q - 1.
    - This relationship is written mathematically as follows −

```
ed = 1 mod (p - 1)(q - 1)
```

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

Example

An example of generating RSA Key pair is given below. For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high.

- Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.
- Select e = 5, which is a valid choice since there is no number that is common factor of 5 and $p - 1q - 1 = 6 \times 12 = 72$, except for 1.
- The pair of numbers n, e = 91, 5 forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.
- Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29.
- Check that the d calculated is correct by computing −

```
de = 29 × 5 = 145 = 1 mod 72
```

- Hence, public key is 91, 5 and private keys is 91, 29.

Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy.

Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n.

RSA Encryption

- Suppose the sender wish to send some text message to someone whose public key is n, e.
- The sender then represents the plaintext as a series of numbers less than n.
- To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

$$C = P^e \bmod n$$

- In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.
- Returning to our Key Generation example with plaintext P = 10, we get ciphertext C −

$$C = 10^5 \bmod 91$$

RSA Decryption

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair n, e has received a ciphertext C.
- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

```
Plaintext = C^d mod n
```

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 −

```
Plaintext = 82^29 mod 91 = 10
```

RSA Analysis

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

- **Encryption Function** − It is considered as a one-way function of converting plaintext into ciphertext and it can be reversed only with the knowledge of private key d.
- **Key Generation** − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe.

The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.