Asymmetric Key Cryptography

DR. VASUDHA ARORA

VASUDHA.ARORA@GDGU.ORG, VASUDHARORA6@GMAIL.COM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GD GOENKA UNIVERSITY, GURUGRAM

Drawbacks of Private Key Cryptography

- traditional **private/secret/single key** cryptography uses **one** key that needs to be shared by both sender and receiver
- if this key is disclosed communications are compromised.
- Difficulties in sharing keys on an open network.
- **symmetric** since parties are equal
- For n parties to communicate with each other, safe communication would require n(n-1)/2. Difficult to play with such a large number of keys to be remembered.
- Secret key is shared and hence does not protect sender from receiver forging a message & claiming it to be sent by sender , i.e no means of authentication provided.

Public (Asymmetric) Key Cryptography

- probably most significant advancement in the more than 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
- complements **rather than** replaces private key cryptography

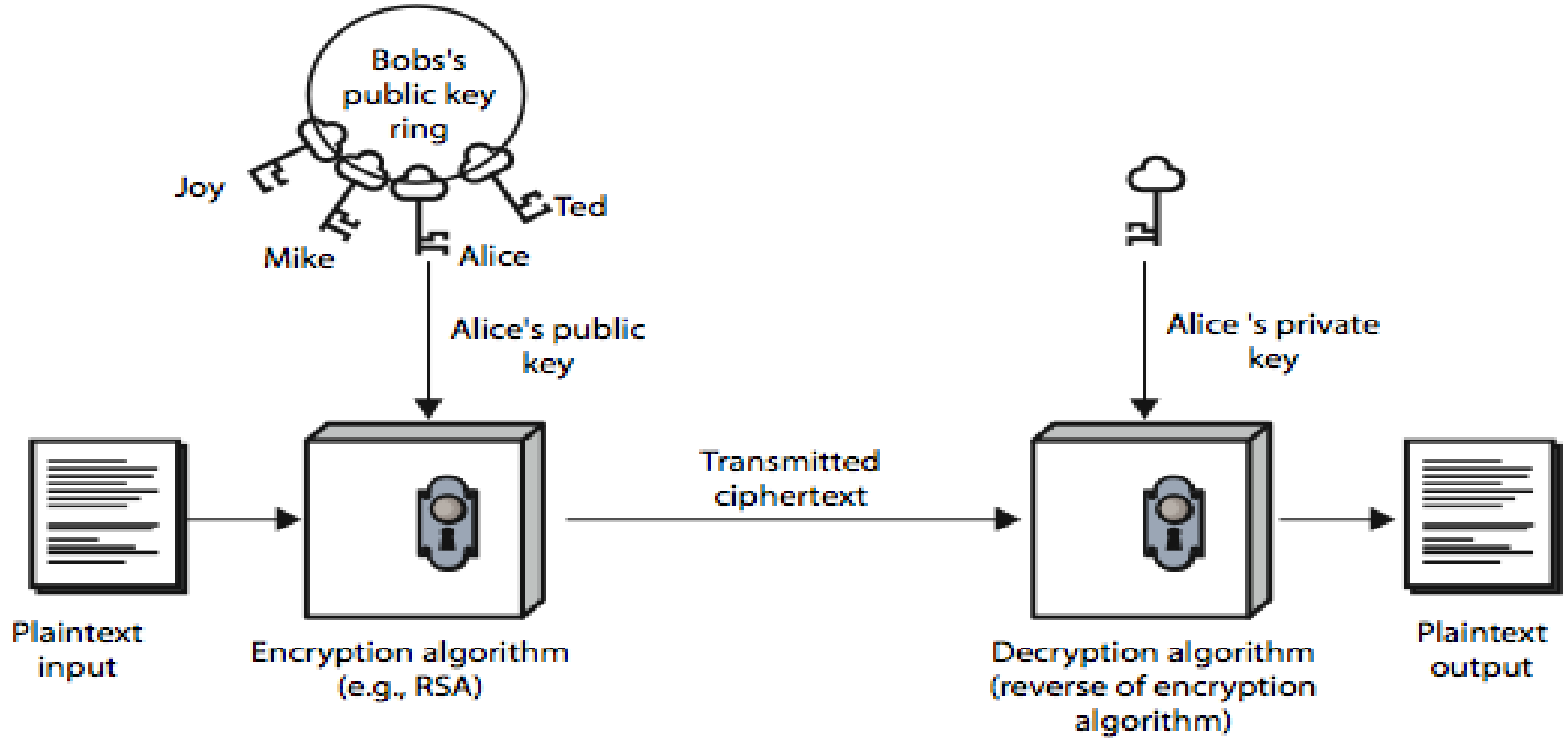
Why Public-Key Cryptography?

- 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
- public invention due to **Whitfield Diffie** & **Martin Hellman** at Stanford University in 1976
  - known earlier in classified community

Public-Key Cryptography

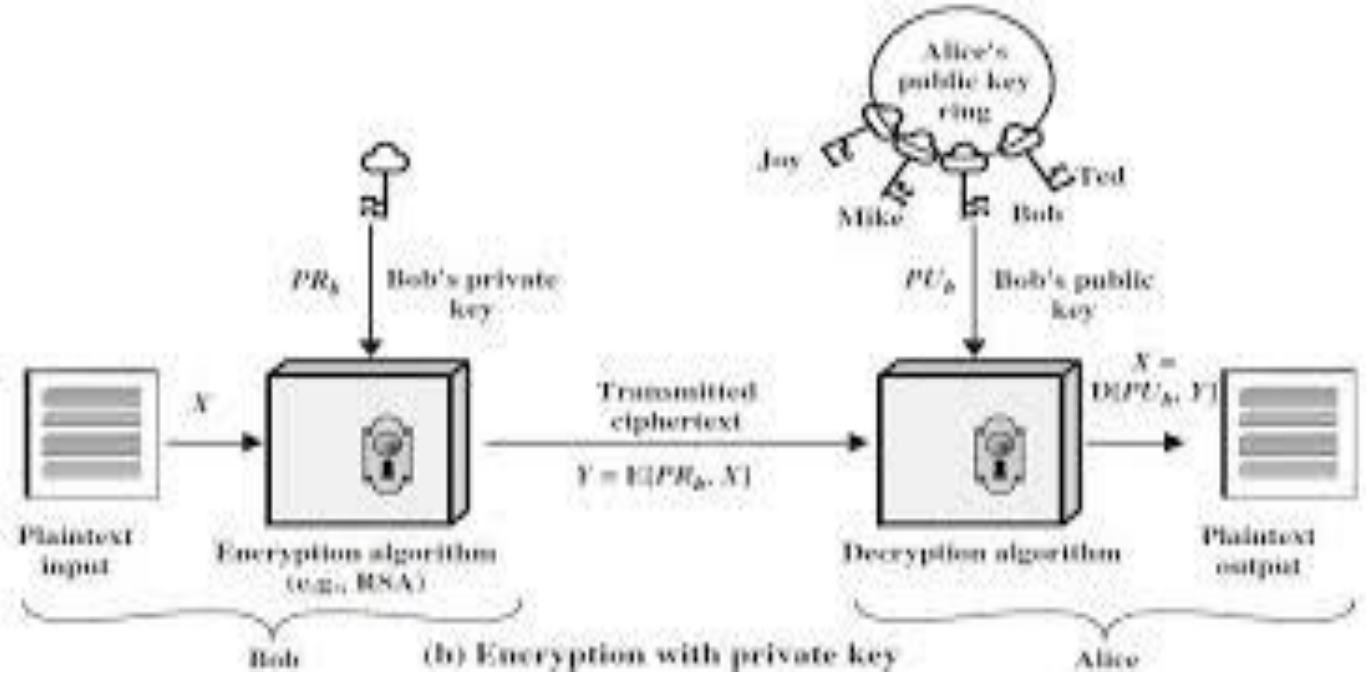
- **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- is **asymmetric** because
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

Public-Key Cryptography



(a) Encryption **with Public Key**

Public-Key Cryptography



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 encrypt/decrypt messages when the relevant (encrypt/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)

Security of Public Key Schemes

- like private key schemes brute force **exhaustive search** attack is always theoretically possible
- but keys used are too large (>512bits)
- security relies on a **large enough** difference in difficulty between **easy** (encrypt/decrypt) and **hard** (cryptanalysis) problems
- more generally the **hard** problem is known, but is made hard enough to be impractical to break
- requires the use of **very large numbers**
- hence is **slow** compared to private key schemes

Public-Key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)
- some algorithms are suitable for all uses, others are specific to only one of the application.

Public-Key Cryptosystem: Secrecy

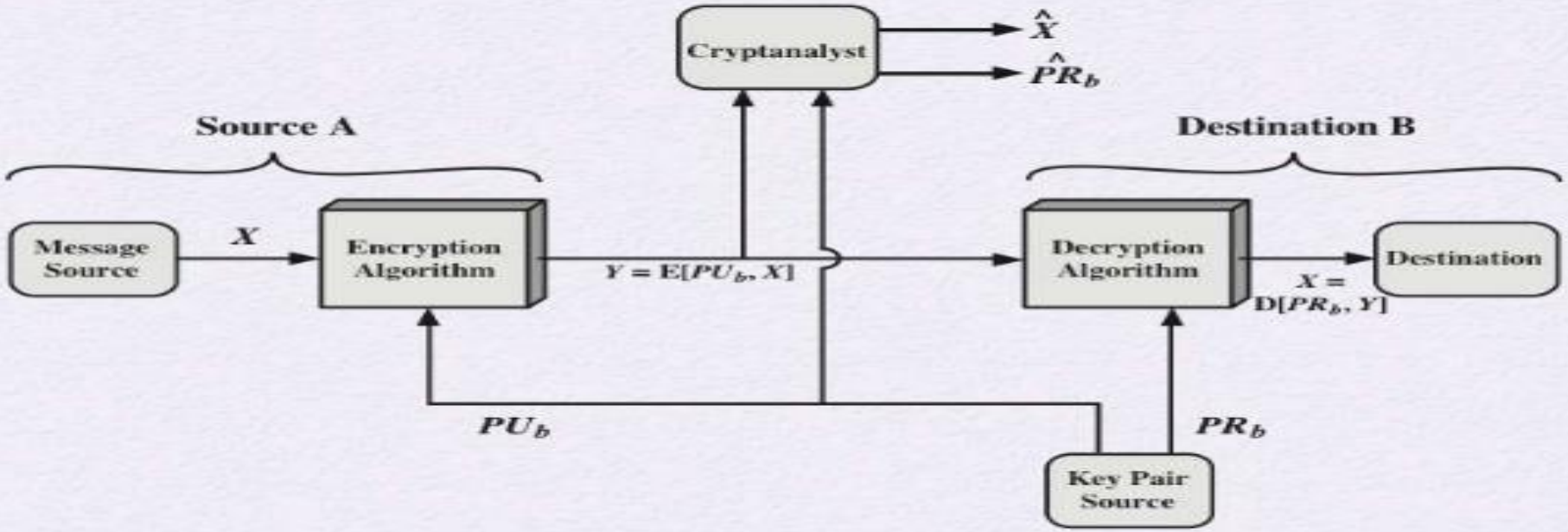


Figure 9.2 Public-Key Cryptosystem: Secrecy

Public-Key Cryptosystem: Authentication

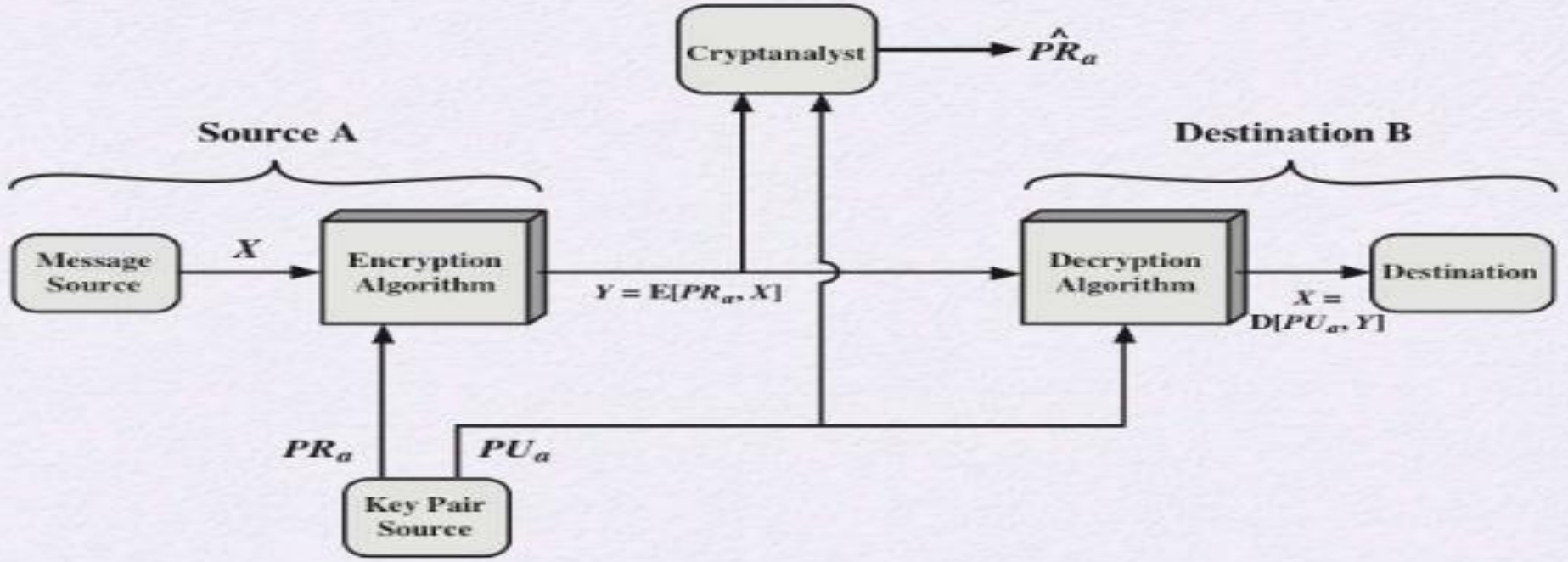


Figure 9.3 Public-Key Cryptosystem: Authentication

Public-Key Cryptosystem: Authentication and Secrecy

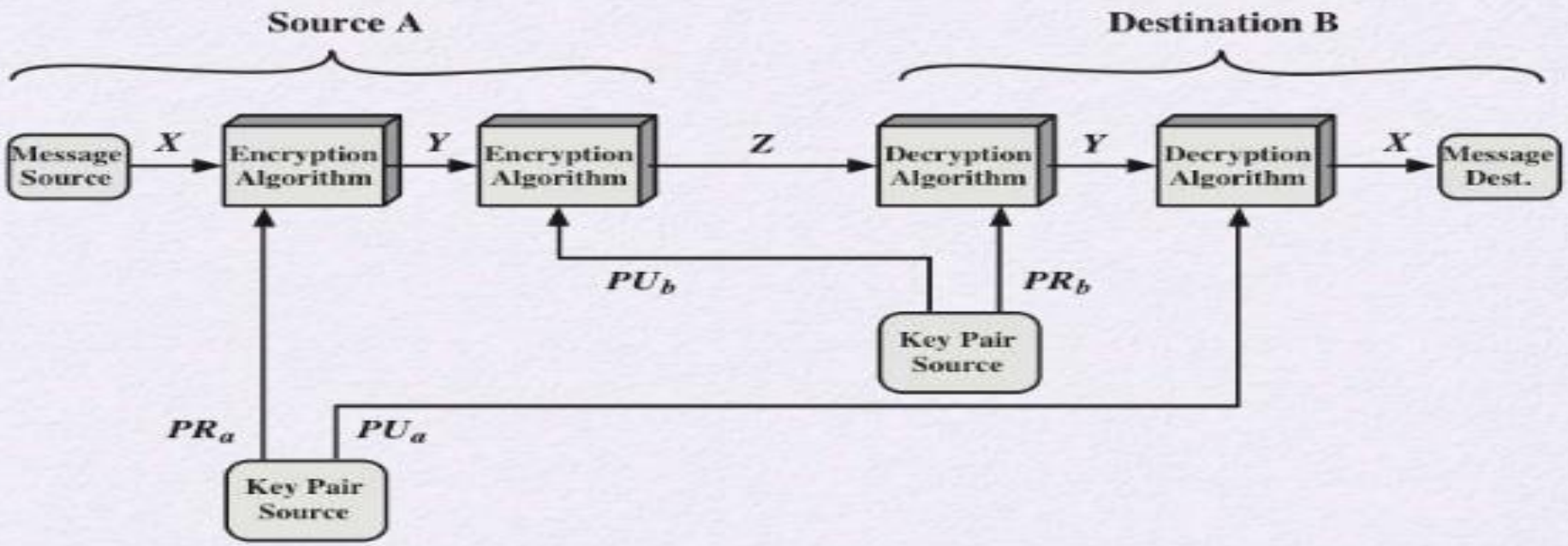


Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

Symmetric v/s Asymmetric

Characteristic	Symmetric Key Cryptography	Asymmetric Key Cryptography
Key used for encryption / decryption	Same key is used for encryption and decryption	One key used for encryption and another, different key is used for decryption
Speed of encryption / decryption	Very fast	Slower
Size of resulting encrypted text	Usually same as or less than the original clear text size	More than the original clear text size
Key agreement / exchange	A big problem	No problem at all
Number of keys required as compared to the number of participants in the message exchange	Equals about the square of the number of participants, so scalability is an issue	Same as the number of participants, so scales up quite well
Usage	Mainly for encryption and decryption (confidentiality), cannot be used for digital signatures (integrity and non-repudiation checks)	Can be used for encryption and decryption (confidentiality) as well as for digital signatures (integrity and non-repudiation checks)

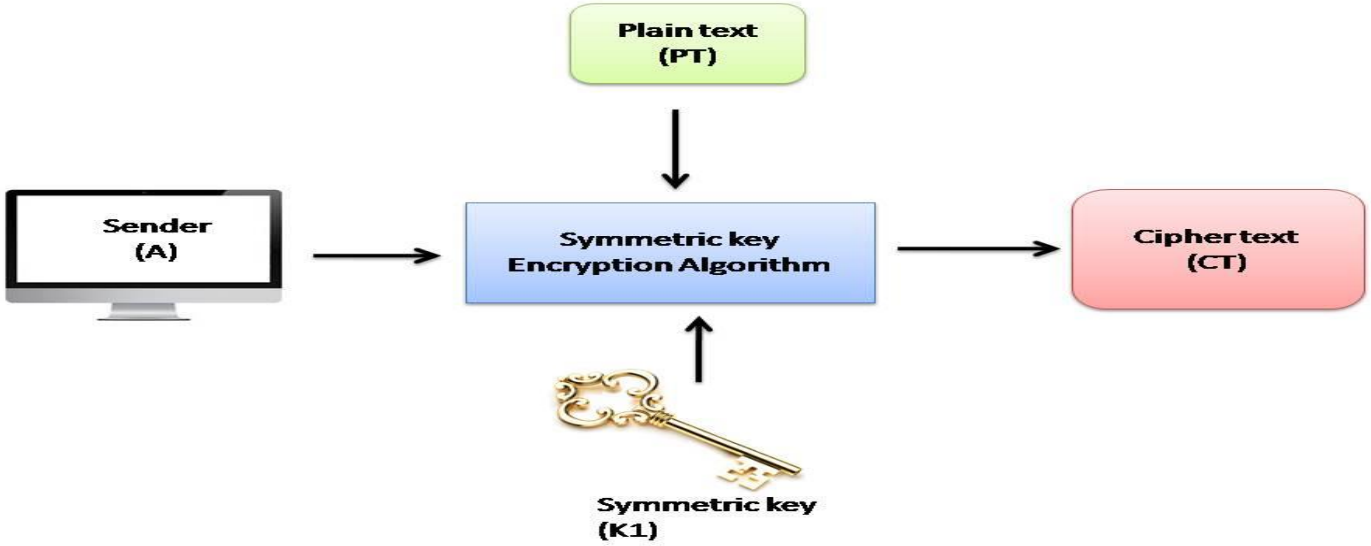
	Symmetric	Asymmetric
Nonrepudiation	No	Nonrepudiation/ Authenticity
Reuse	Requires new keys for each new group that wants to communicate	No need to regenerate keys unless they are compromised
Effectiveness	1,000-10,000 times faster than asymmetric algorithms	Slow
Scalability	Not scalable	Easily scalable
Distribution	Requires parties to securely exchange private keys	Public keys are freely available/private keys do not need to be shared

Disadvantages of Public Key Cryptography

- Very Slow Encryption/Decryption as compared to Symmetric key cryptography.
  - Almost 100 to 1000 times slower.
- Size of encrypted data limited by performance considerations
  - Not suitable for encrypting large amounts of data.

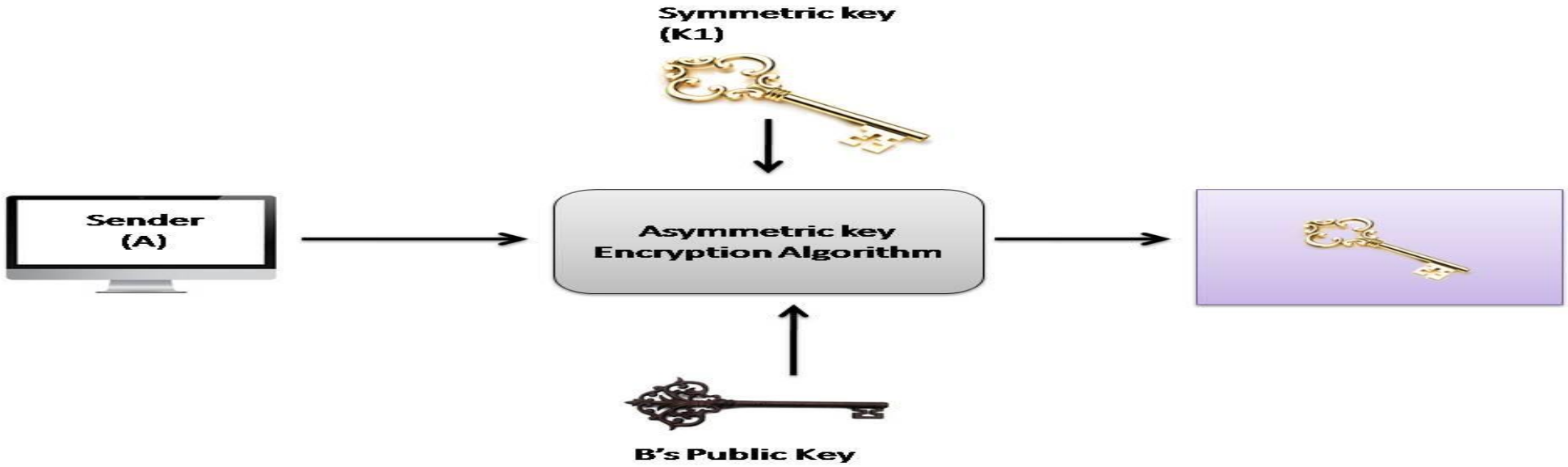
Symmetric and Asymmetric key cryptography together

- Indeed, in practice, symmetric-key cryptography and asymmetric-key cryptography are combined to have a very efficient security solution. The way it works is as follows, assuming that A is the sender of message and B is its receiver.
- A's computer encrypts the original plain-text message (PT) with the help of a standard symmetric key cryptography algorithm, such AES, DES, IDEA or RC5, etc. this produces a cipher-text message (CT) . The key used in this operation (K1) is called one-time symmetric key, as it is used once and then discarded.



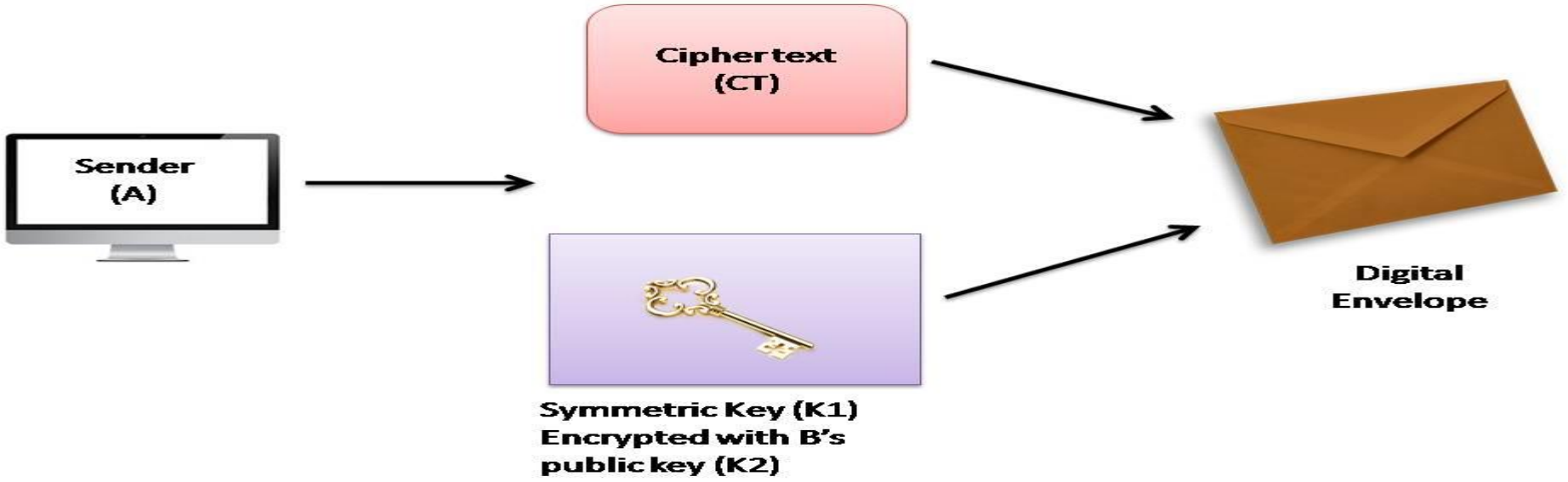
Symmetric and Asymmetric key cryptography together

- Does this not again lead us to the key-exchange problem? Well, a novel concept is used now.
- A now takes the one-time symmetric key of step 1 (i.e. K1), and encrypts K1 with B's public key (K2). This process is called **key wrapping** of the symmetric key, We can imagine it as that the symmetric key K1 goes inside a logical box, which is sealed by B's public key (i.e. K2).



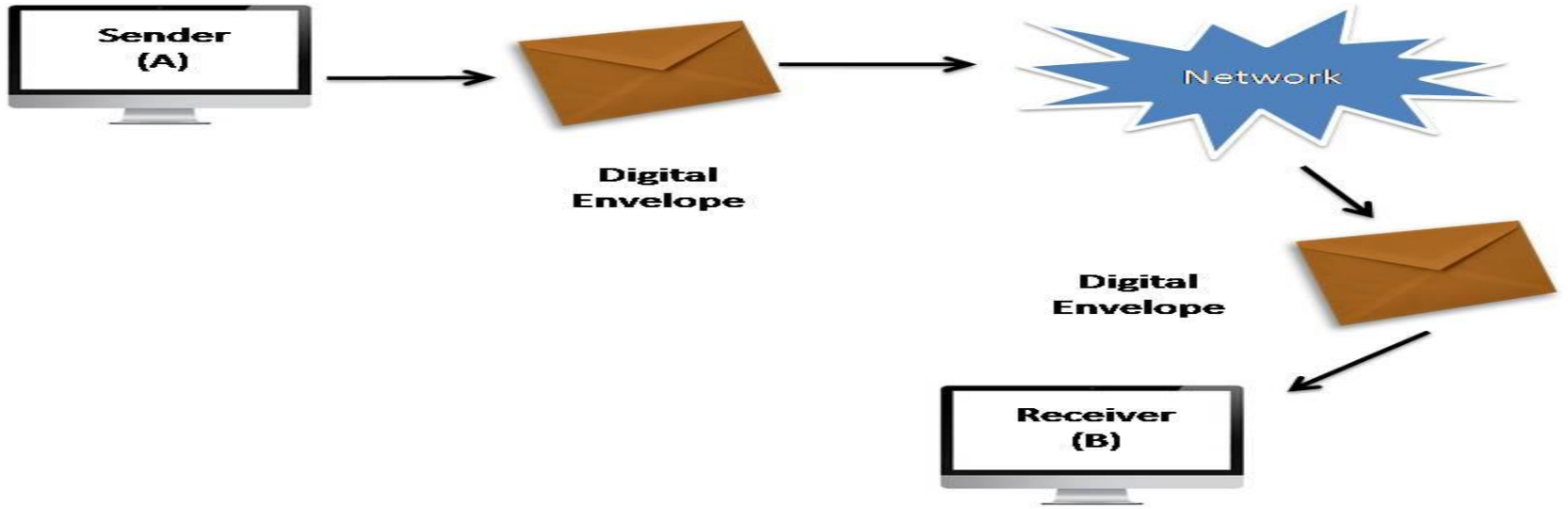
Symmetric and Asymmetric key cryptography together

- Now, A puts the cipher text CT1 and the encrypted symmetric key together inside a digital envelope.



Symmetric and Asymmetric key cryptography together

- The sender (A) now sends the digital envelope [which contains the cipher text (CT) and the onetime symmetric key (K1) encrypted with B's public key, (K2)] to B using the underlying transport mechanism (network). This is shown in fig .we do not show the contents of the envelope, and assume that the envelope contains the two entities, as discussed.

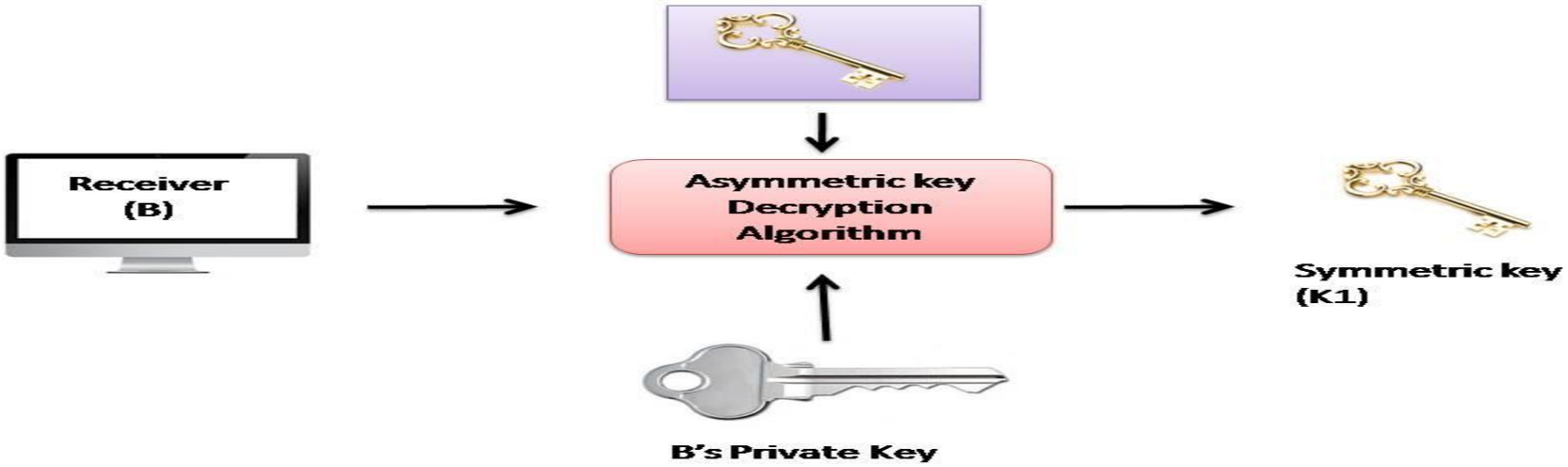


Symmetric and Asymmetric key cryptography together

- B receives digital envelope and opens it . After B opens this digital envelope, he gets 2 things first is cipher text (CT) and another one is the one-time session key (K1) which is encrypted using B's public key (K2).

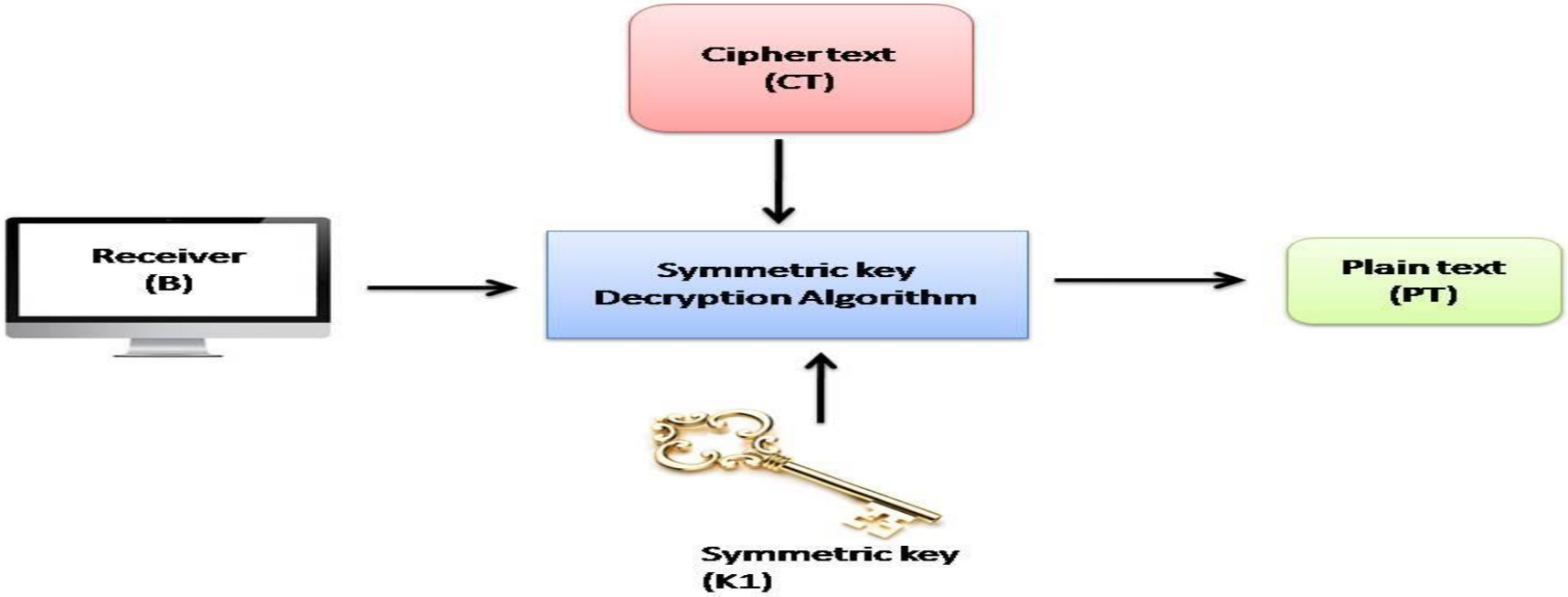
Symmetric and Asymmetric key cryptography together

- B now uses the same asymmetric-key algorithm as was used by A and her private key (K3) to decrypt (i.e. open up) the logical box that contains the symmetric key (K1), which was encrypted with B's public key (K2). This the output of the process is the one-time symmetric key K1.



Symmetric and Asymmetric key cryptography together

- Finally, B applies the same symmetric-key algorithm as was used by A, and the symmetric key K1 to decrypt the cipher text (C1). This process yields the original plain text (PT), as shown in fig. below.



Reasons for Efficiency

- first
- second
- Third