Cryptography

- Cryptography transforms readable messages into an unintelligible form and then later reverses the process
- It can be used to send sensitive data securely over an untrusted network
- It uses authentication and encryption methods

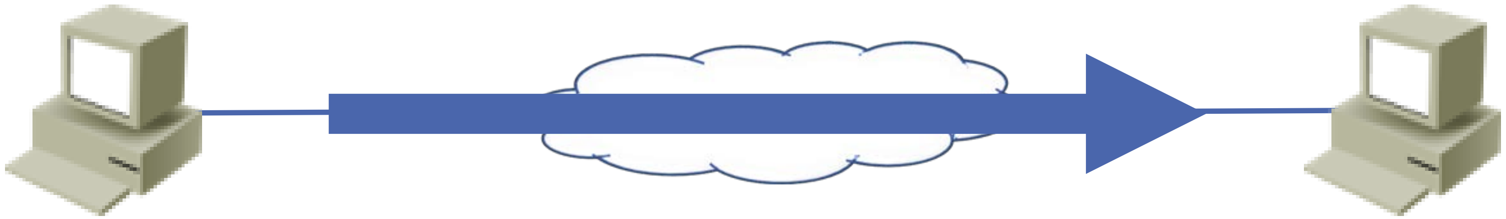
Cryptography Services

- Cryptography provides these services to data:
- Authenticity (proof of source)
- Confidentiality (privacy and secrecy)
- Integrity (has not changed in transit)
- Non-repudiation (non-deniability)

Symmetric Encryption

- With symmetric encryption, the same shared key both encrypts and decrypts the data
- The shared key is known by both the sender and receiver and must be kept secret
- Fast
- Used for large transmissions (eg email, secure web traffic, IPsec)
- Algorithms include DES, 3DES, AES, SEAL

Symmetric Encryption - Confidentiality



Data: Hello

Encrypts with
Shared Key: 123

&^%$#&*&

&^%$#&*&

&^%$#&*&

Decrypts with
Shared Key: 123

Data: Hello

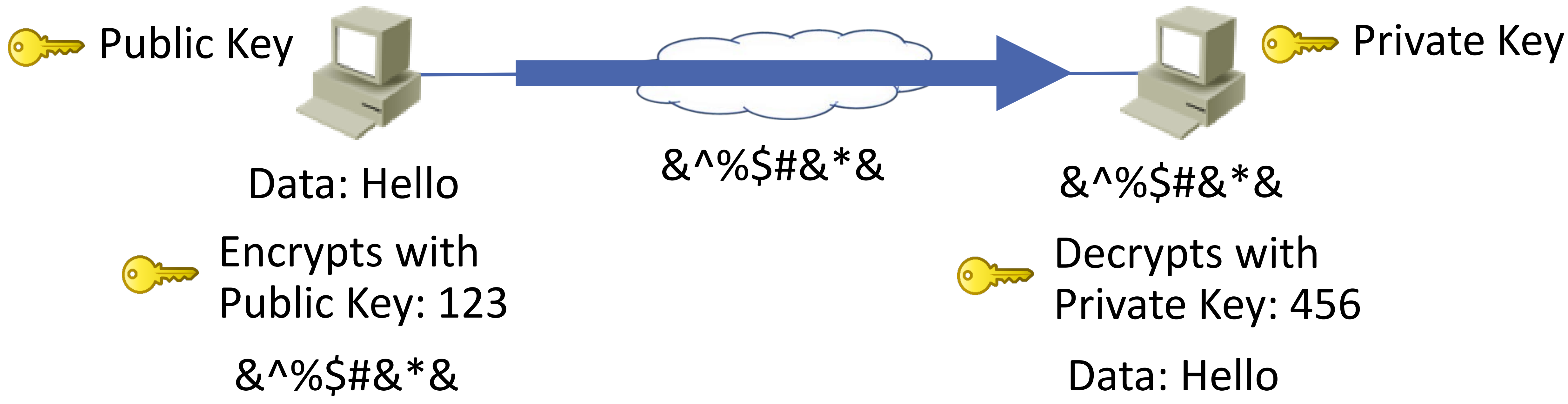
Asymmetric Encryption

- Asymmetric encryption uses private and public key pairs
- Data encrypted with the public key can only be decrypted with the private key, and vice versa
- Data encrypted with the public key **cannot** be decrypted with the public key
- Only the private key must be kept secret

Asymmetric Encryption (Cont.)

- The public key can be available in the public domain
- Slow
- Used for small transmissions (symmetric key exchange, digital signatures)
- Algorithms include: RSA, ECDSA

Asymmetric Encryption - Confidentiality



Public Key

Data: Hello

Encrypts with
Public Key: 123

&^%$#&*&

&^%$#&*&

Private Key

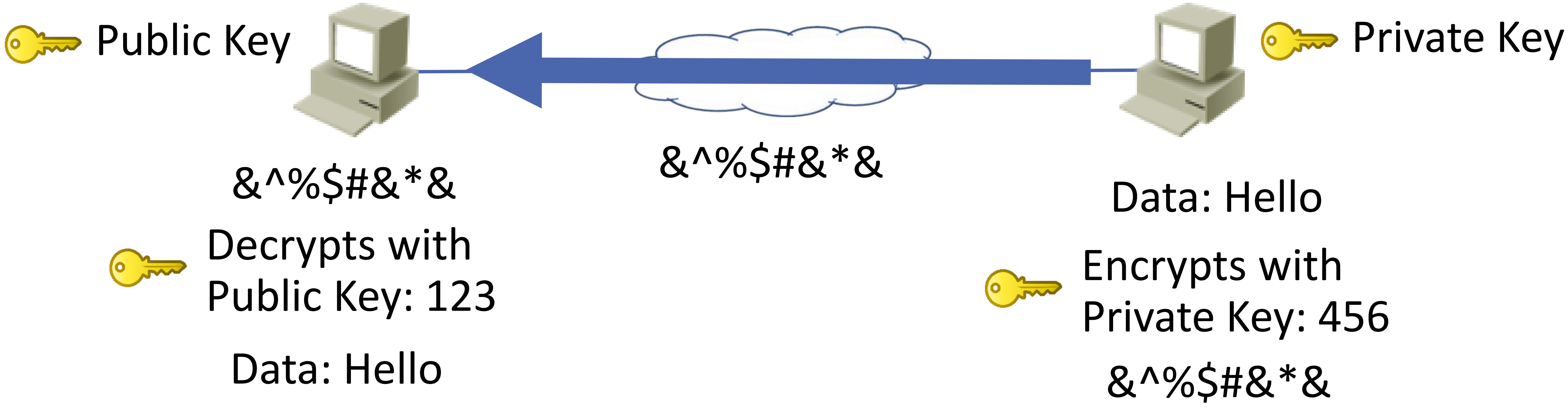
&^%$#&*&

Decrypts with
Private Key: 456

Data: Hello

- This allows anybody to send data securely to the host with the private key
- It is the only one with the private key so only one who can read the message
- Other hosts with the public key **cannot** read the message

Asymmetric Encryption – Authenticity and Non-Repudiation



Public Key

Private Key

&^%$#&*&

&^%$#&*&

Decrypts with
Public Key: 123

Data: Hello

Data: Hello

Encrypts with
Private Key: 456

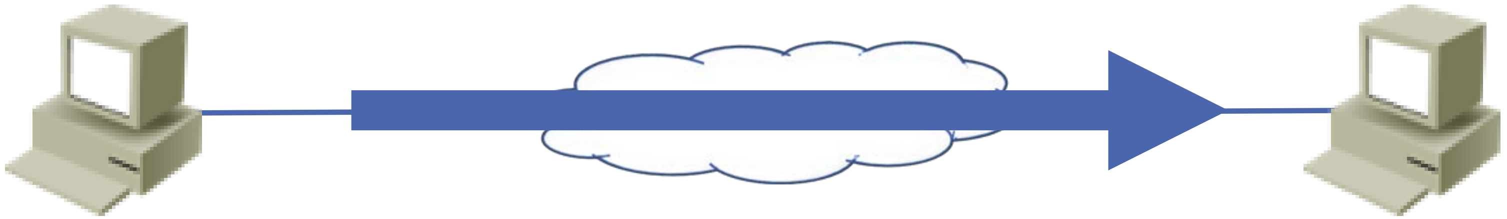
&^%$#&*&

- This provides authenticity of the host with the private key
- All receivers need to know what is the genuine public key

HMAC Hash-Based Message Authentication Codes

- HMAC codes provide data integrity
- The sender creates a hash value from the data to be sent using a symmetric key
- The hash value is appended to the data
- The receiver hashes the data with the same shared key
- If the hash values are the same the data has not been altered in transit
- Used for large transmissions (eg email, secure web traffic, IPsec)
- Algorithms include: MD5, SHA

HMAC - Integrity



Data: Hello

Hashes with
Shared Key: 123

Hash: &^%$

Hello

Hash: &^%$

Data: Hello

Hash: &^%$

Hashes data with
Shared Key: 123

Hash: &^%$

Key Distribution

- Cryptography can be used to send sensitive data securely over an untrusted network
- Symmetric key encryption is used for bulk data transmissions
- Each side needs to know the shared key
- This leads to a key distribution problem

Key Distribution (Cont.)

- When you buy something online, you want your credit card details to be encrypted over the Internet
- The online store can’t send you the shared key over the same Internet channel - it’s not secure
- It’s not practical for them to phone the customer every time someone wants to make a purchase

Public Key Infrastructure PKI

- PKI solves the secure key distribution problem
- It uses a trusted introducer (the Certificate Authority) for the two parties who need secure communication
- Both parties need to trust the Certificate Authority