



Encrypted Sockets

Part 2 – Cipher Suites

Barak Gonen

Part 2 – Cipher Suites

- ▶ Hashing
- ▶ Data Integrity
- ▶ Encryption
 - Asymmetric
 - Symmetric
- ▶ Authentication



Hashing

- ▶ “One directional” function
 - Easy to calculate the hashed result from the source
 - Very difficult to calculate source from hashed results
 - Example: sum of digits
123459 \rightarrow 24

Hashing

- ▶ “Good” hash characteristics:
 - Fixed length result
 - Small change in source, yields big change in result
 - No two sources yield same result

“Improved” Hash

- ▶ Improved hash:
 - Source X 1234
 - Square the result
 - Pick digits in indexes 4–9
 - Square the result
 - Pick digits in indexes 4–8
- ▶ How many sources may be (at most) before the results repeat?

Source	Result
99	57569
100	83553
101	64178

Common Hash Algorithms

- ▶ MD5 – 128 bits, declared unsafe in 2008
- ▶ SHA1 – 160 bits, declared unsafe in 2017
- ▶ SHA2 – 256, 384, 512 bits

- ▶ Exercise:
 - Create two text files. First file should store “hello cool cyber class”. Second file should store the same, plus a dot in the end. Use Powershell to find their MD5 and SHA256

Get-FileHash filename -algorithm MD5/SHA256

- Do the same in python

MD5 hash using encode()

```
import hashlib

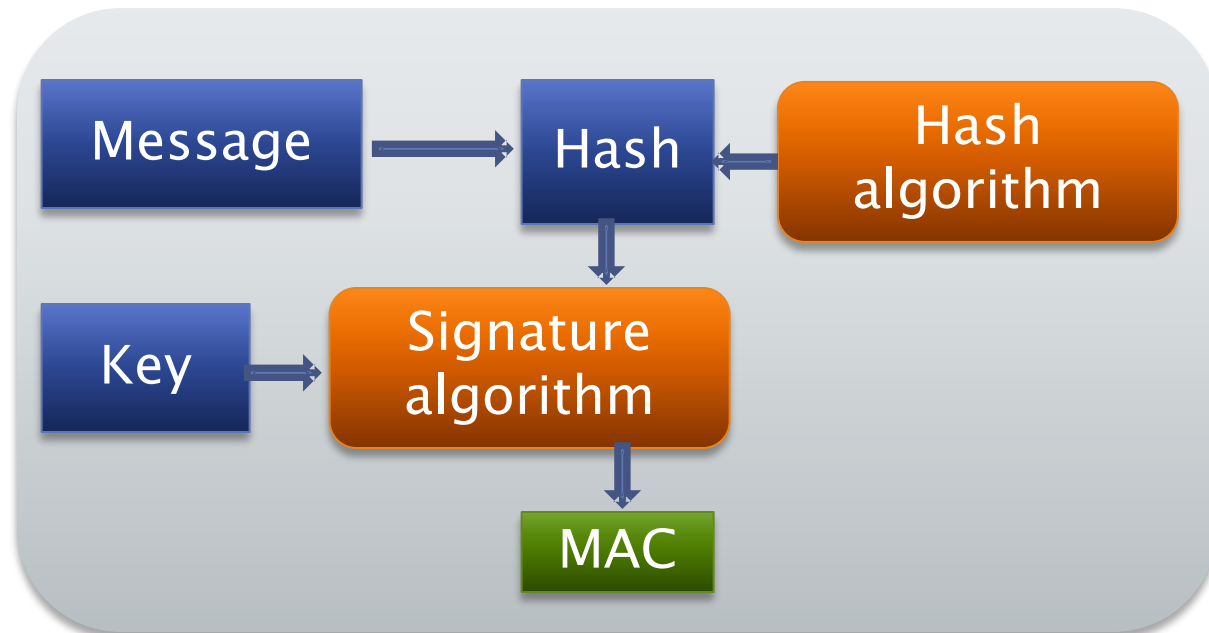
# initializing string
str1hash = "Hello cool cyber class"
str2hash = "Hello cool cyber class."

# encoding string using encode() and then sending to md5()
result = hashlib.md5(str1hash.encode())
result2 = hashlib.md5(str2hash.encode())
print(result)
print(result2)

# printing the equivalent hexadecimal value.
print("The hexadecimal equivalent of hash is : ", result.hexdigest())
print("The hexadecimal equivalent of hash is : ", result2.hexdigest())
```

Data Integrity

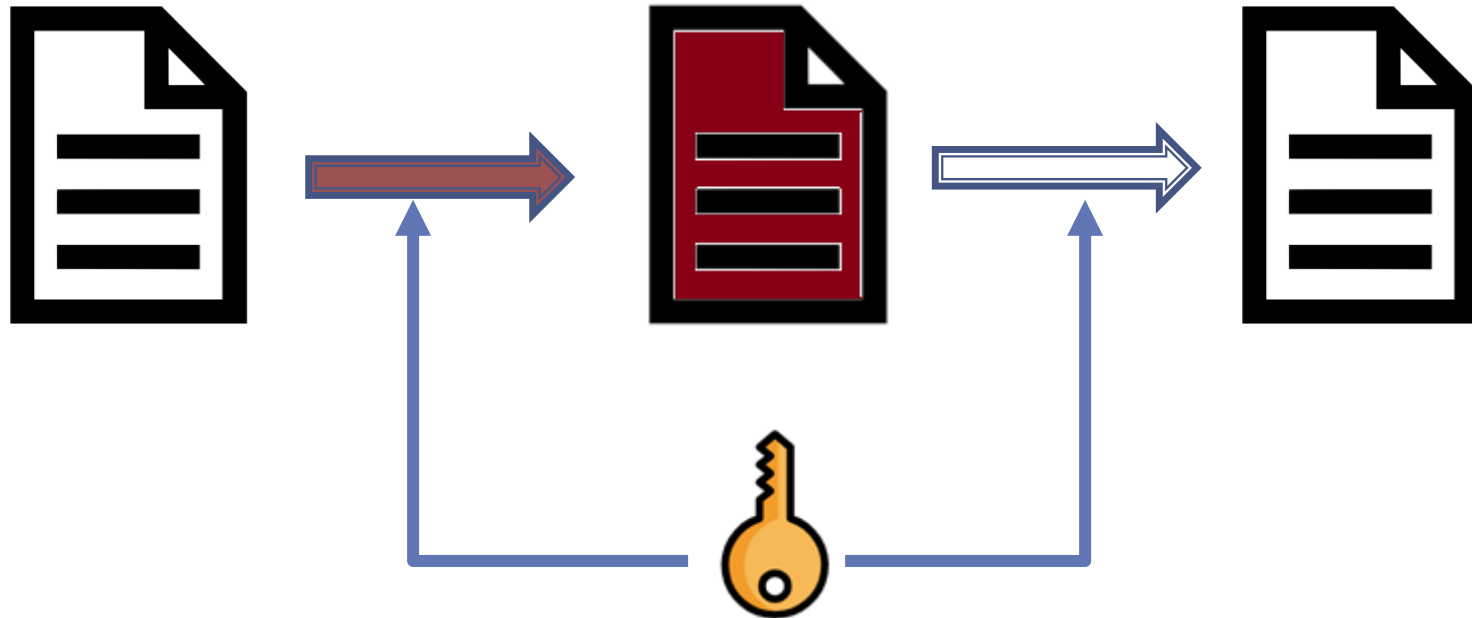
- ▶ Method to ensure data was not changed
- ▶ Message Authentication Code (MAC)
- ▶ The key is shared only by sender and recipient
- ▶ Recipient compares hash to calculated hash



Key Based Encryption

- ▶ Public algorithm
- ▶ Secret key
- ▶ Encryption types:
 - Symmetric
 - Asymmetric

Symmetric Encryption



- ▶ Same key used for encryption / decryption
 - Shift cypher
 - XOR function

XOR Encryption – Example

Data: 1001 0011

Key: 0101 0100

1001 0011 xor
0101 0100

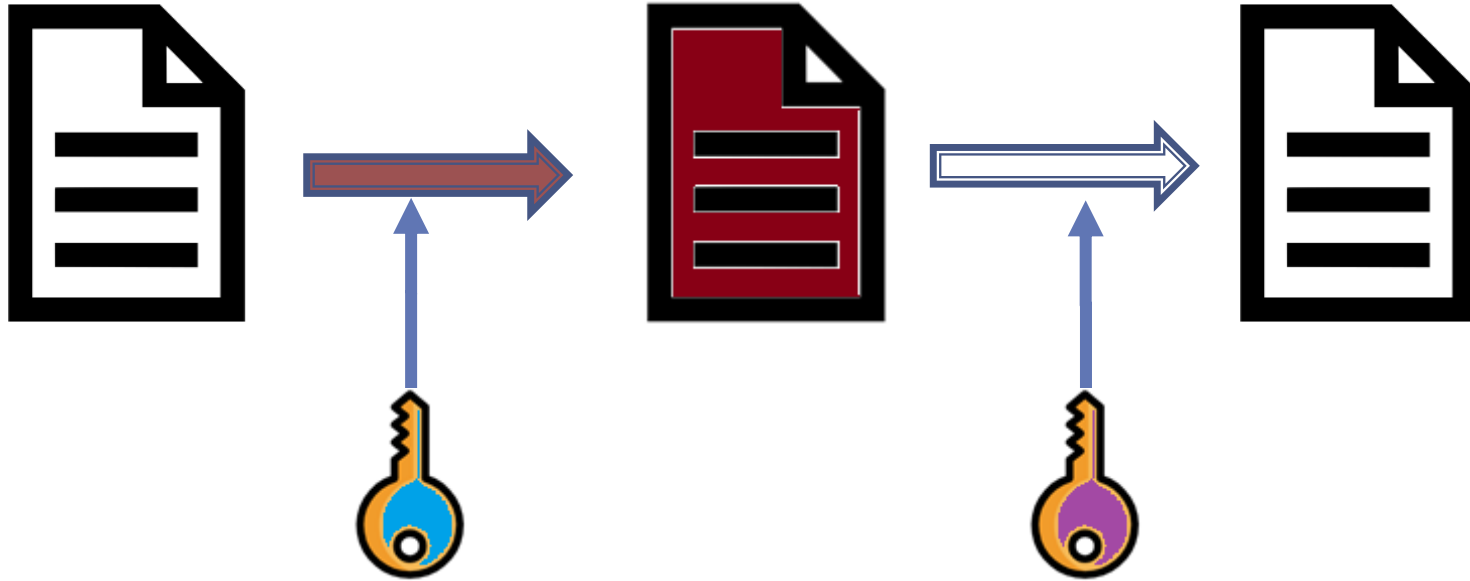
1100 0111 – encrypted

1100 0111 xor
0101 0100

1001 0011

- ▶ The key is required to decrypt

Asymmetric Encryption



- ▶ Different keys are used to encrypt / decrypt
 - Keys come in pairs
 - Only the pair key can decrypt
 - Order of keys does not matter

RSA – Asymmetric Cipher

- ▶ Rivest, Shamir Edleman 1977
- ▶ Used for exchanging symmetric keys
 - Idea: public key, private key
- ▶ Other usages:
 - Integrity– digital signatures
 - Encryption (slow, but can be used)

RSA

- ▶ Pick two prime numbers – P, Q
- ▶ Calculate $P \times Q$
- ▶ Calculate Totient $T = (P-1) \times (Q-1)$
 - totient (plural totients) (mathematics) The number of positive integers not greater than a specified integer that are relatively prime to it.
- ▶ Pick public key E
 - Prime
 - Smaller than T
 - $T \bmod E$ is not zero
- ▶ Pick a private key D
 - Condition: $D \times E \bmod T = 1$
- ▶ Exercise:
 - Let $P=17$, $Q=23$
 - Let public key E be 113
 - Find the matching private key ?

Encryption / Decryption

- ▶ Cipher = $(\text{Plain}^E) \bmod N$
- ▶ Plain = $(\text{Cipher}^D) \bmod N$

Other side is given:

- N (“Modulus”)
- E (“Exponent”)
- In order to break the private key, N must be broken to P and Q

$$P = 17$$

$$Q = 23$$

$$N = 391 \quad (P \times Q)$$

$$T = 352 \quad (P-1)(Q-1)$$

$$E = 113 \quad (\text{Public})$$

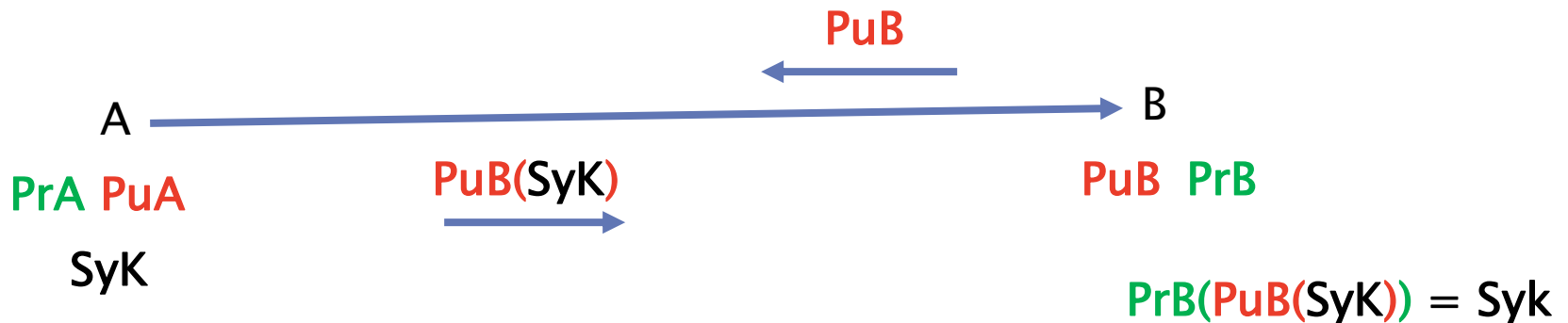
$$D = 81 \quad (\text{Private})$$

Pros and Cons

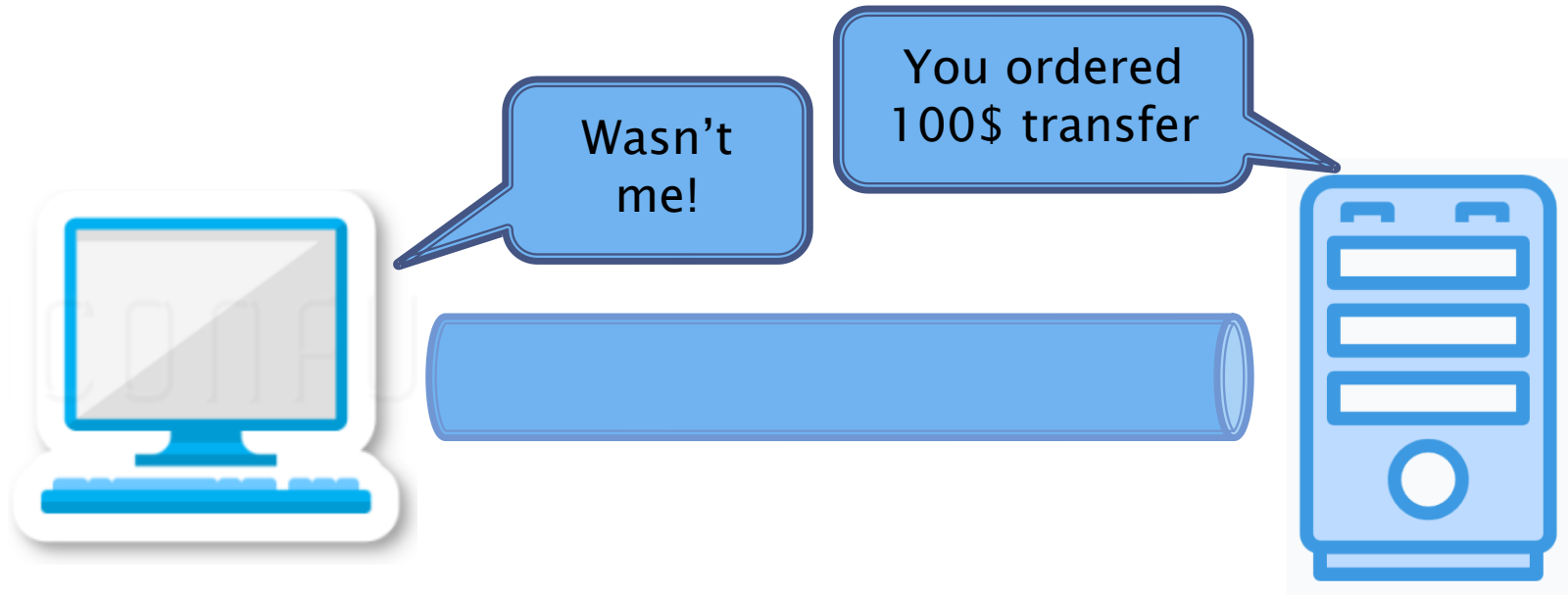
- ▶ Symmetric encryption:
 - Faster
 - Key must be shared between sender–recipient
 - How can both sides share the key?
- ▶ Asymmetric encryption:
 - Slow
 - No need to share keys

Exchange of Symmetric Keys

- ▶ Common usage of RSA:
 - Alice creates a key for symmetric encryption
 - Alice encrypts the key with Bob's public key
 - Bob decrypts the key with his pair, private key
 - Now Alice and Bob share the same key



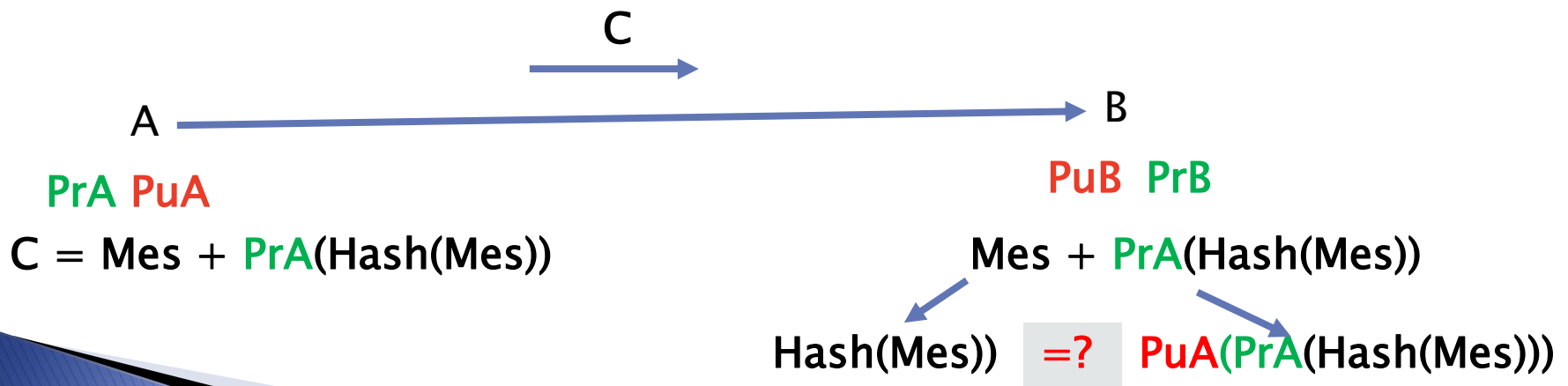
Non-Repudiation



- ▶ Sender cannot deny sending a message
- ▶ Logically, a result of:
 - No one is able to impersonate the sender
 - No one is able to change the message
- ▶ How can that be achieved?

Non-Repudiation

- ▶ Alice calculates hash of message
- ▶ Alice signs the message: encrypts hash with private key
- ▶ Alice's public key is known, so Bob can decrypt the digital signature
- ▶ Bob calculates the hash and compares the result to the decrypted hash
- ▶ If both are equal, Alice cannot deny sending the message
- ▶ ... Unless her private key is broken



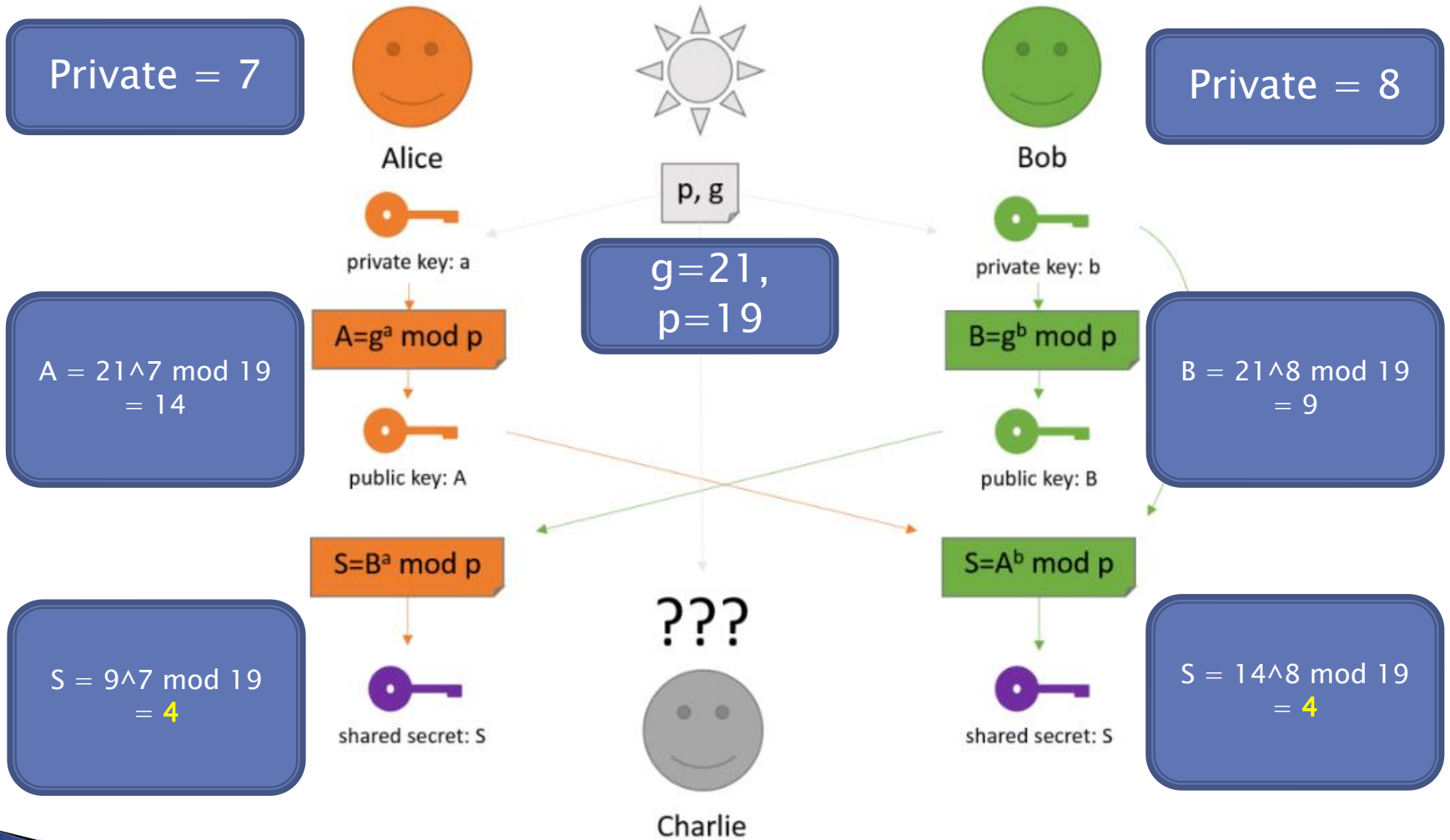
RSA Summary

- ▶ Encryption – possible, slow
- ▶ Symmetric key exchange
- ▶ Authentication = Non-Repudiation + Integrity

Diffie–Hellman Key Exchange

- ▶ Usage – creating a shared secret
 - Not an encryption algorithm
 - DH is commonly used to coordinate symmetric key

DH Algorithm



Symmetric Encryptions in TLS

- ▶ Lesson focus:

- AES
- GCM vs CBC

- ▶ CHACHA20
- ▶ AES 256 GCM
- ▶ AES 128 GCM
- ▶ AES 256 CBC
- ▶ AES 128 CBC
- ▶ 3DES CBC
- ▶ RC4 128
- ▶ DES CBC

AES

- ▶ Advanced Encryption System 1998
- ▶ Block cipher
 - Input 128 bits into 4x4 byte matrix
- ▶ Repeat:
 - XOR with key
 - Substitution cipher
 - Switch rows
 - Switch columns

AES -cont.

- ▶ Problem 1: same input will always result in same output



Source :Wikipedia, Block Cipher

AES – cont.

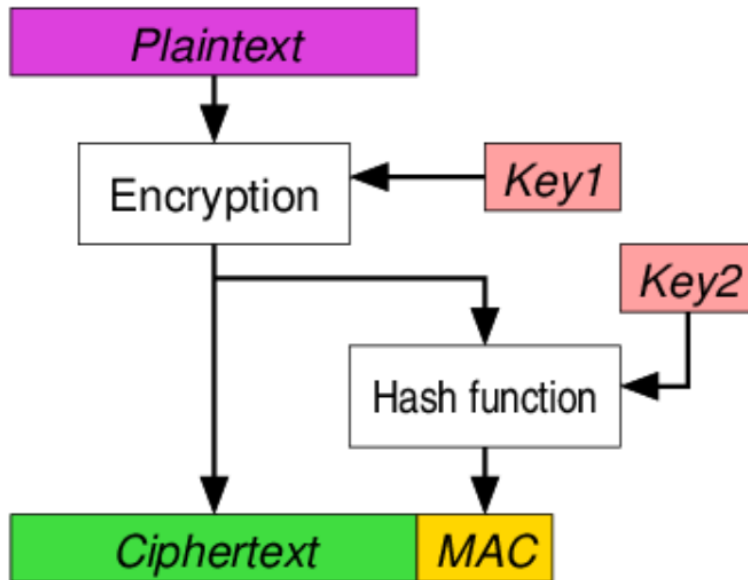
► Solutions:

- CBC – Cipher Block Chaining – last bytes of previous block are used for generating new key for next block
 - Blocks can't be decrypted in parallel
- CTR – Counter Mode. Initial key, every block uses last key + counter
 - Parallel decryption possible

AES – cont.

- ▶ Problem 2: CTR does not include –
 - Integrity
 - Authentication

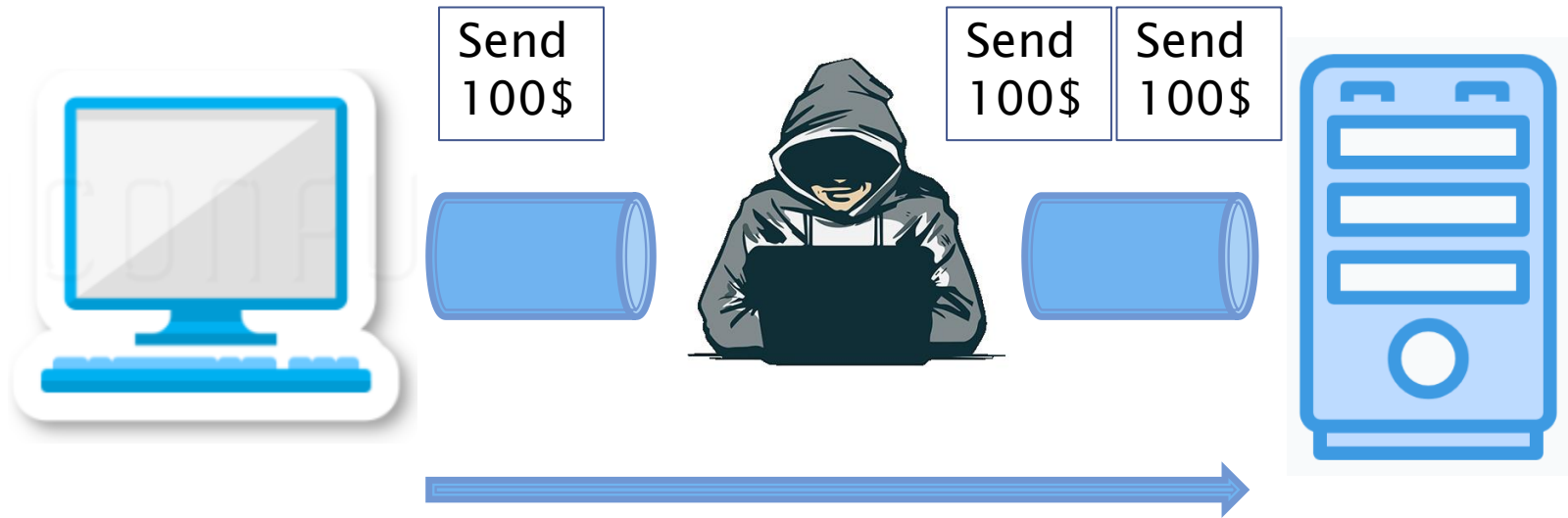
AES – cont.



Source :Wikipedia,
Authenticated Encryption

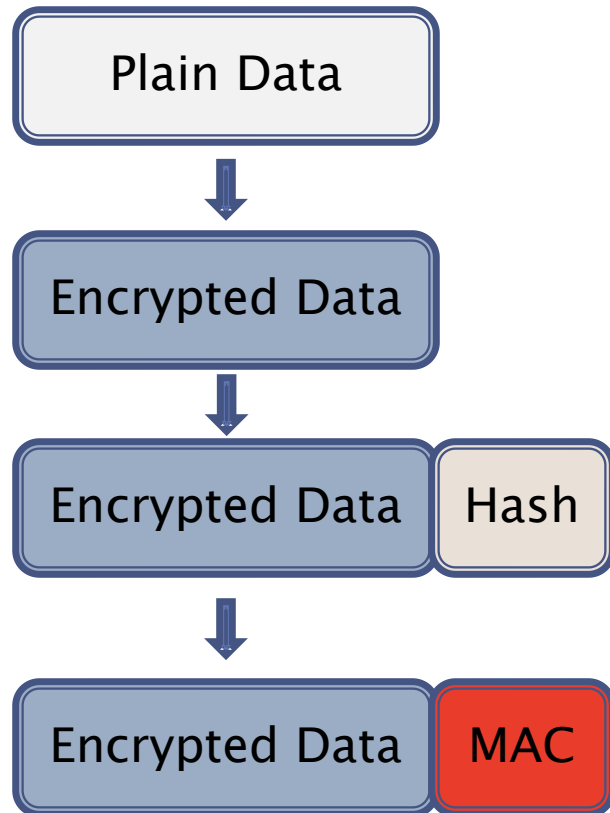
- ▶ Solution: combine encryption & authentication
- ▶ AEAD – Authenticated Encryption with Associated Data
- ▶ GCM – Galois Counter Mode
 - GCM = AEAD + Counter mode

Anti-replay



- ▶ The copied message will be rejected
 - Every message will be decrypted with a different key
 - The MAC will not be correct

Cipher Suites



TLS Cipher Suites

- ▶ Client – Server should coordinate 4 issues
- ▶ Format:
- ▶ TLS_x_y_with_z_w
 - X – Key Exchange
 - Y – Authentication
 - Z – Encryption
 - W – Hashing

Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

Summary

- ▶ The scheme allows:
 - Sharing a key with other side
 - Know that the message was sent by the key owner
 - Know that the message was not changed on way
- ▶ But what if the keys were exchanged with an imposter?

