COMP435: *SECURITY CONCEPTS!*

Lecture 11: Pitfalls of Authentication, Start Systems

tinyurl.com/comp435-fa25

SOME MORE TERMS

Terminology

- Protocol
- Unilateral Authentication
- Mutual Authentication
- Key Establishment
- Session Key

Protocol

Def’n: well defined sequence of messages between communicating parties

Unilateral Authentication

Def’n: one communicating party verifies the identity of the other party

E.g., A user’s browser authenticates the identity of the bank’s server

Mutual Authentication

Def’n: each communicating party verifies the identity of the other

E.g., Two end-user devices authenticate to each other

Session Key

Def’n: a key used for short-term purposes

E.g., A new session key is established each time you log in to your bank.

Principles for Cryptographic Protocols

1. Every message should say what it means--its interpretation should depend only on its content

2. The conditions for a message to be acted upon should be clearly set out so that someone reviewing a design may see whether they are acceptable or not

--Abadi and Needham, 1995

https://www.andrew.cmu.edu/course/18-330/2020f/reading/PrudentEngineeringProtocols.pdf

PITFALLS IN AUTHENTICATION PROTOCOLS

Insecure

1.

I'm Purple, my password is S

2.

This is Purple, password: S

Replay Attack

1.

I'm Purple, $\mathrm{H}(S)$

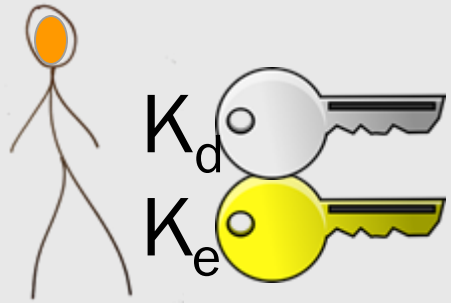
2.

I'm Purple, $\mathrm{H}(S)$

Hybrid Encryption: Naïve Implmelenation

1.

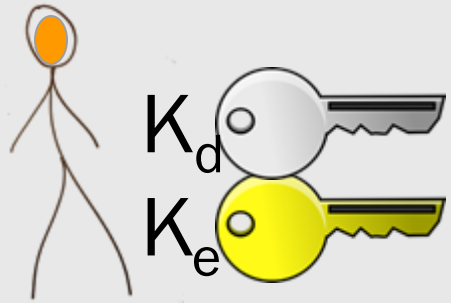
I'm Purple. Let's use: RSA-$\mathtt{Enc}_{\mathtt{Ke}}(\mathtt{S})$



Hybrid Encryption: Naïve Implmelenation

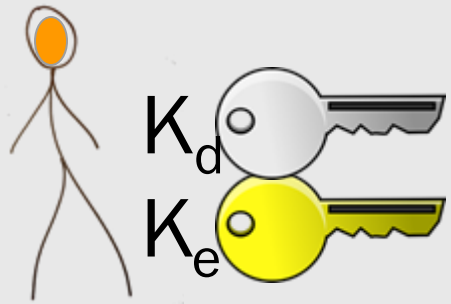
1.

I'm Purple. Let's use: $Enc_{Ke}(S)$



2.

I'm Purple. Let's use: $Enc_{Ke}(S)$



Susceptible to replay attack!

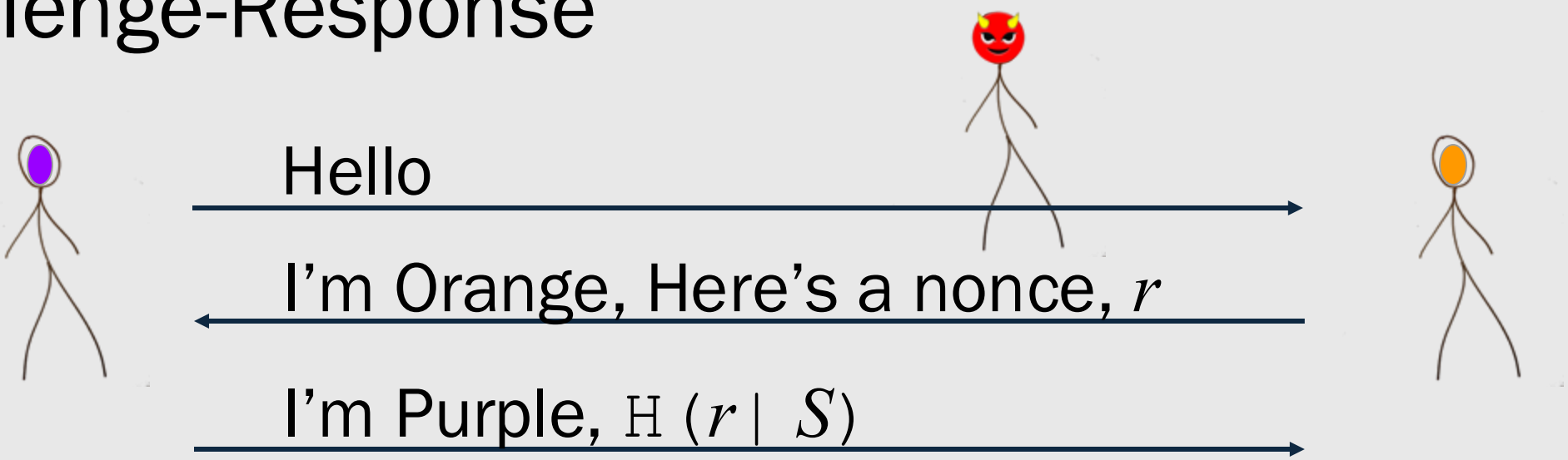
Nonce

Def’n: a number used only once as part of a cryptographic protocol

- random number used to provide freshness, message continuity
- sequence number to demonstrate uniqueness
- timestamp to demonstrate timeliness

E.g., IV used in CBC, OFB, Ctr modes of block ciphers

Challenge-Response



Challenge-Response

Hello

I'm Orange, Here's a nonce, r

I'm Purple, $\mathrm{H}(r \mid S)$

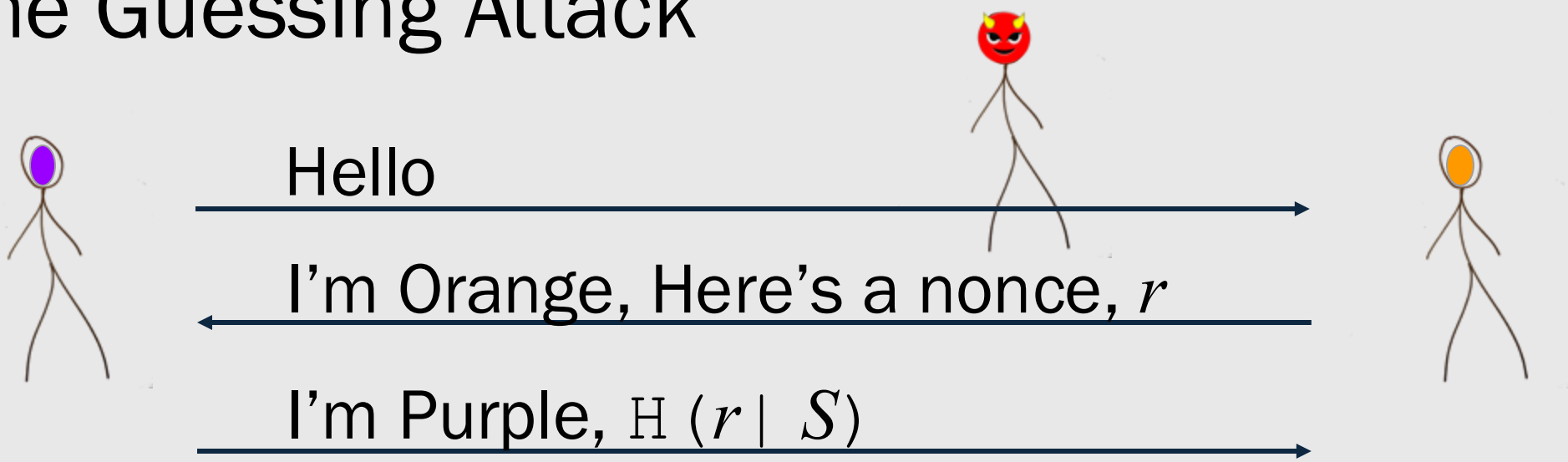
Hello

I'm Orange, Here's a nonce, r'

I'm Purple, $\mathrm{H}(r \mid S)$

Replay attack won't work!

Offline Guessing Attack



$\mathrm{H}(\mathrm{r},\text{"12345"}) \stackrel{?}{=} \mathrm{H}(r, S)$

$\mathrm{H}(\mathrm{r},\text{"qwerty"}) \stackrel{?}{=} \mathrm{H}(r, S)$

...

Reflection Attack

I'm Orange. Prove you are Purple.

Here's a nonce, r

I'm Purple. Prove you are Orange. Here's a nonce, r

I'm Orange. $\mathrm{H}(r \mid S)$

I'm purple. $\mathrm{H}(r \mid S)$

Machine in the Middle Attack (MitM) (or Adversary in the Middle (AitM))

Def’n: two parties, believing that they are communicating with each other, are in fact communicating with an attacker

Adversary-in-the-Middle attack on DH

No authentication ☹

a

$A := g^a \bmod p$

b

$B := g^b \bmod p$

D

B

$K_{ad} := D^a \bmod p$

K_{ad}

K_{cb}

$K_{cb} := C^b \bmod p$

TRUST

Trusted vs. Trustworthy

- **Trust**:
  - *relied upon to meet its security specifications*
  - *Trust is about one system depending on another system*
- **Trustworthy**:
  - *deserving of trust; meets its security specifications*
  - *Trustworthiness is about a system's security posture*

Trusted Computing Base (TCB)

Def'n: those components upon which the security of the system relies

"Trusted system: a subsystem that is permitted to violate some global security policy"

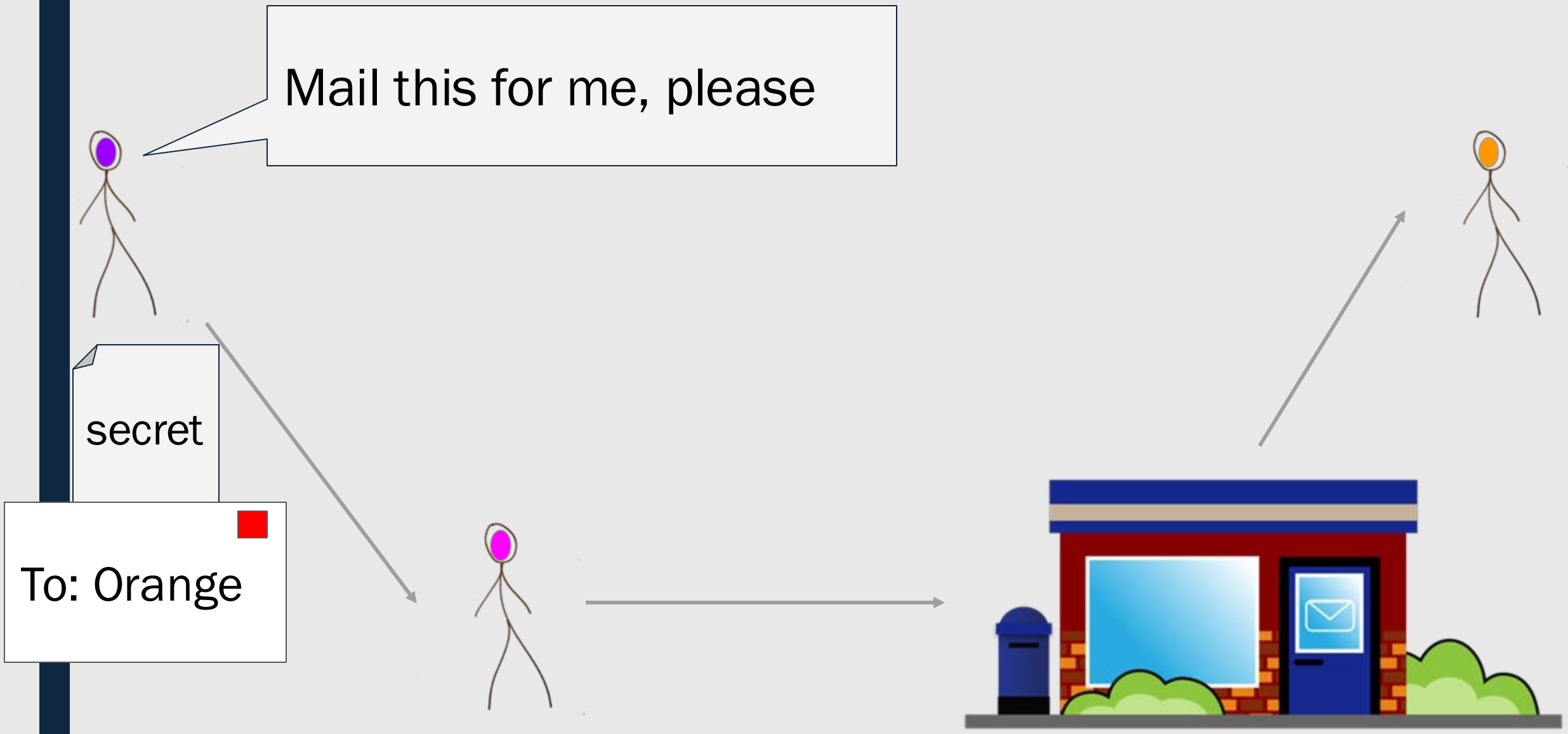
--Goguen and Meseguer

Mail this for me, please

secret

Trustworthy

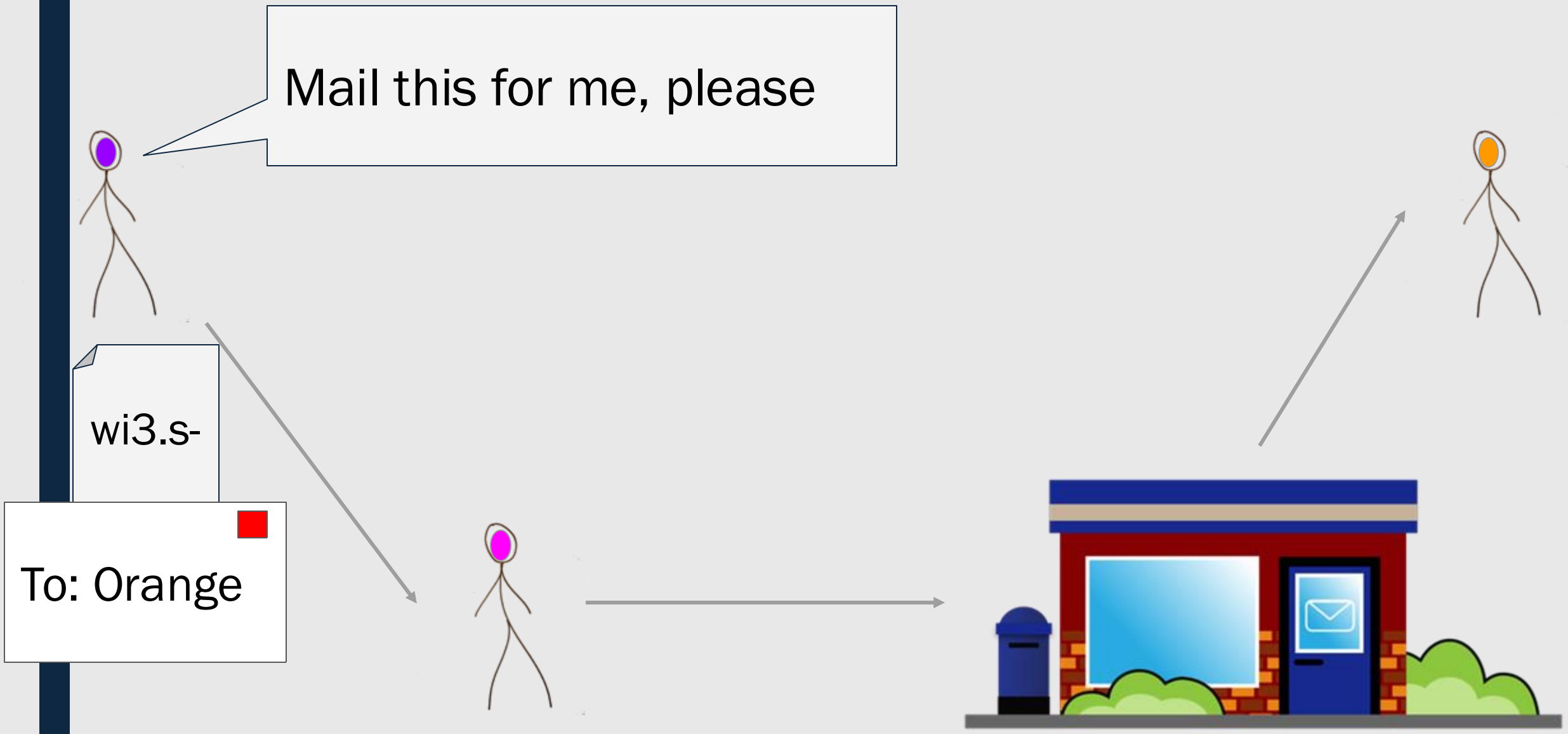
- secure against intentional, malicious actions
- secure against accidental actions
- secure against 3rd party actions


Mail this for me, please
secret
To: Orange

Honest, but Curious

Def’n: an adversary that tries to glean information while adhering to the protocol

E.g., In an honest-but-curious threat model, the TA does not open the sealed envelope


Mail this for me, please
wi3.s-
To: Orange

WS Qs 1-2

TRUSTED SYSTEMS

Trusted Systems Standards

- Orange Book
  - *official name: the Trusted Computer System Evaluation Criteria*
  - *Developed in 1970s by US DoD.*
  - *6 rankings, coupled security features w/ assurance features*

- Common Criteria
  - *Official name: Common Criteria for Information Technology Security Evaluation*
  - *Developed in 1990s by group of International Governmental Agencies*
  - *7 assurance levs, decouple security features from assurance requirements*

Trusted Systems’ Process

- specification
- design process
- evaluation

Trusted Systems' Process

- specification
- design process
- evaluation

defines the desired security policy and functionality

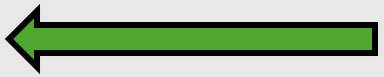
Trusted Systems' Process

- specification
- design process
- evaluation

1. model
2. design
3. implement

Trusted Systems' Process

- specification
- design process
- evaluation



- testing
- red-, blue-team exercises
- formal verification

Features of a Trusted System

- TCB
- trusted path
- secure start-up
- object sanitization
- auditing

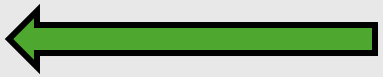
Features of a Trusted System

- TCB
- trusted path
- secure start-up
- object sanitization
- auditing

should be small and well defined

Features of a Trusted System

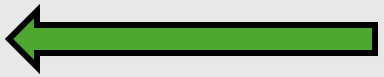
- TCB
- trusted path
- secure start-up
- object sanitization
- auditing



authenticates the OS to the user

Features of a Trusted System

- TCB
- trusted path
- secure start-up
- object sanitization
- auditing



starts in a known good state

Features of a Trusted System

- TCB
- trusted path
- secure start-up
- object sanitization
- auditing

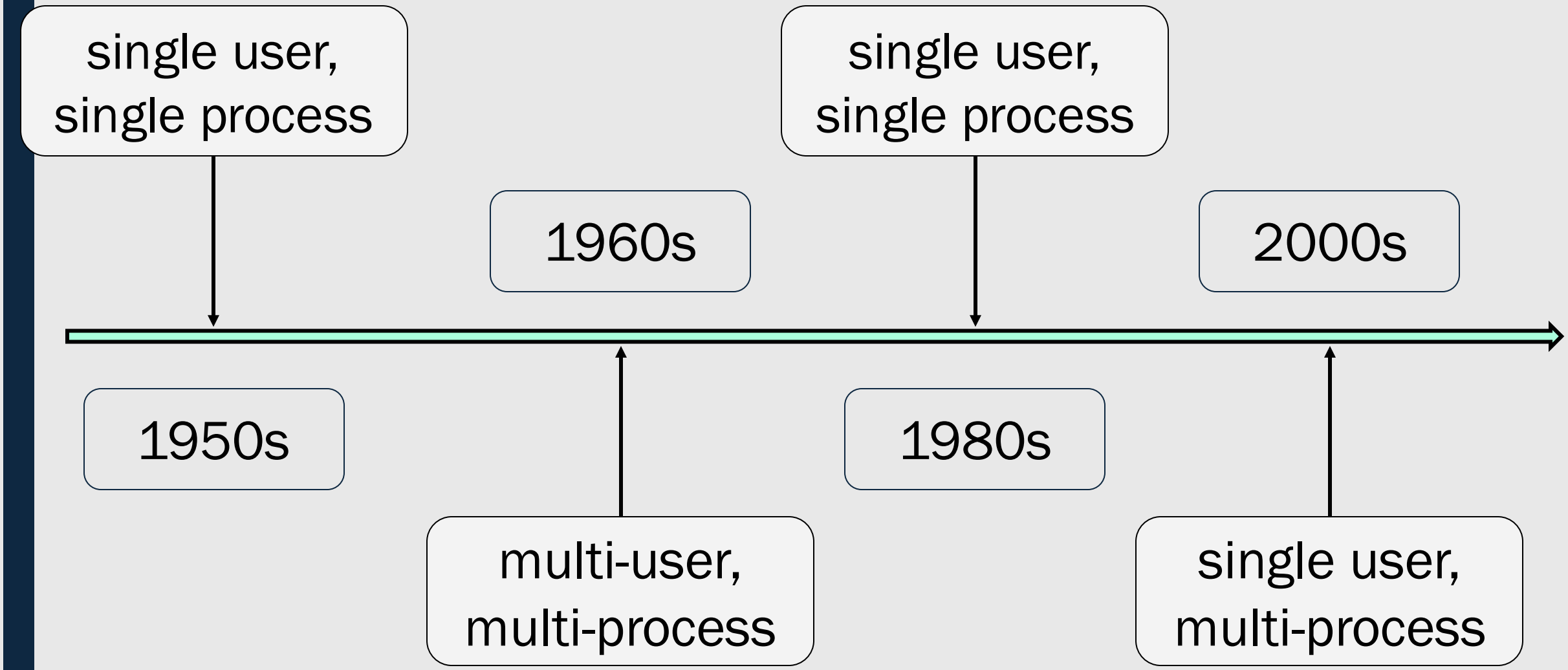
no object reuse
(free your memory!)

Features of a Trusted System

- TCB
- trusted path
- secure start-up
- object sanitization
- auditing

audit log tracks any changes

OPERATING SYSTEMS


single user,
single process
1960s
single user,
single process
2000s
1950s
multi-user,
multi-process
1980s
single user,
multi-process

OS Responsibilities

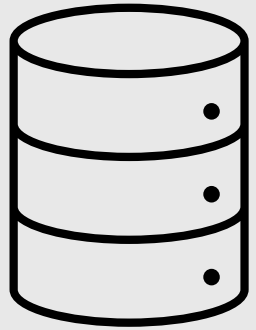
- manage resources
- provide interface to HW
- provide user authentication
- mediate interprocess communication
- protect itself

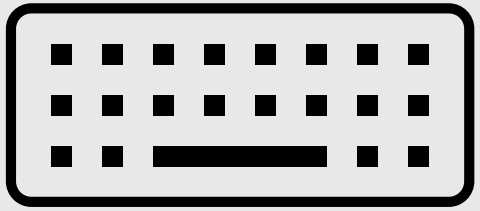
Manage Resources

- allocation
- sharing
- protection

Provide Interface to Hardware

- memory
- file system
- device I/O



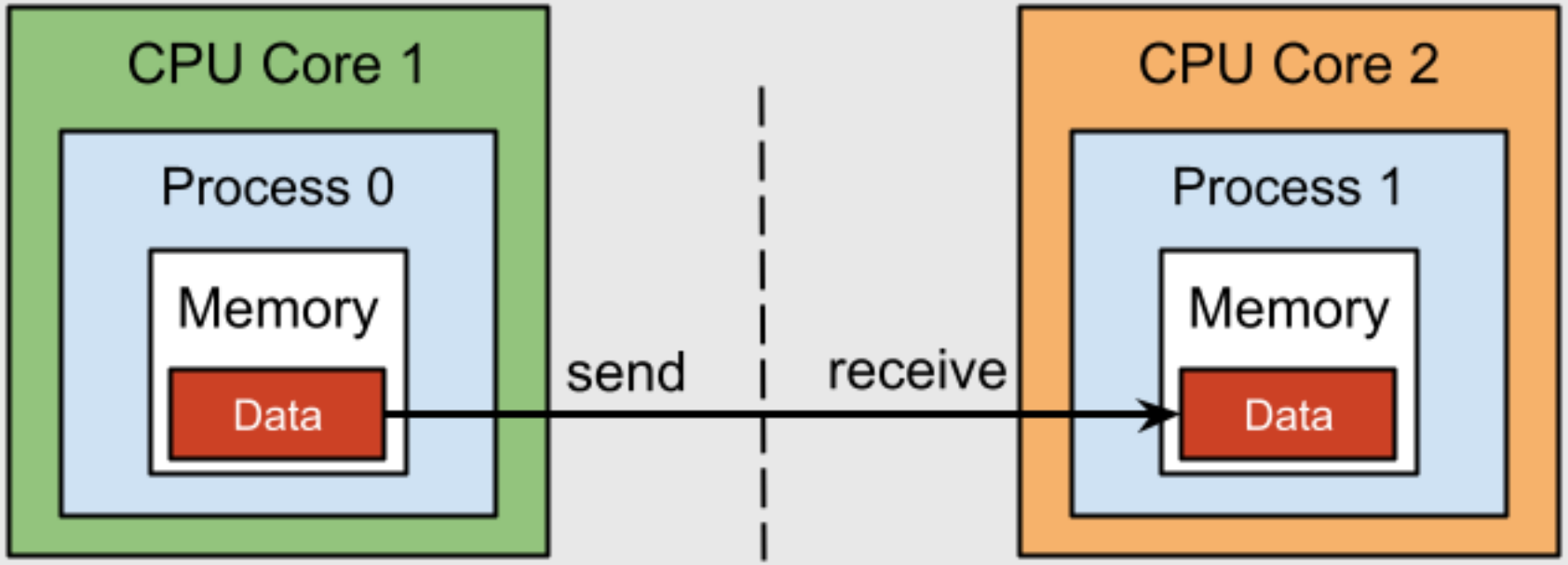


Provide User Authentication

- user accounts
- roles
- passwords

Mediate Interprocess Communication

- shared memory
- message passing



Protect the Operating System

- data
- code
- permissions

RACE CONDITIONS

Race Condition

Def’n: Concurrent access of a resource is not serializable

Occurs when:

- Two or more processes have access to **the same resource**
- The final state of the resource depends on the **relative timing** of the **two processes**

ATM

Bank

balance = 900

withdraw $100

get_balance()

balance = 900

$100

set_balance(800)

balance = 800

ATM

Bank

ATM

balance = 900

w/d $100

w/d $50

get_bal()

get_bal()

bal = 900

bal = 900

$100

$50

set_bal(800)

set_bal(850)

balance = ??

ATM

Bank

ATM

balance = 900

w/d $100

get_bal()

get_bal()

w/d $50

bal = 900

bal = 900

$100

$50

set_bal(800)

set_bal(850)

balance = ??

```
filename = “tmpname”;


// Write to the file
fd = open(filename, O_CREATE|O_RDWR);
```

```
filename = “tmpname”;
           symlink(“/etc/passwd”, “tmpname”);

// Write to the file
fd = open(filename, O_CREATE|O_RDWR);
```

Program A: Privileged Program!

```
1.  filename = "tmpname";
    // Write to the file
2.  fd = open(filename,O_CREATE|O_RDWR);
```

Program B: Attacker!

```
1.  symlink("/etc/passwd", "tmpname");
```

Problem: TOCTTOU
Race condition!