CS 3650 – Computer Systems
Spring 2024
Peter Desnoyers

Lecture 25, Tue Apr 9 2024

Security, access control, etc.

goals: {confidentiality, integrity} × {operations, data}

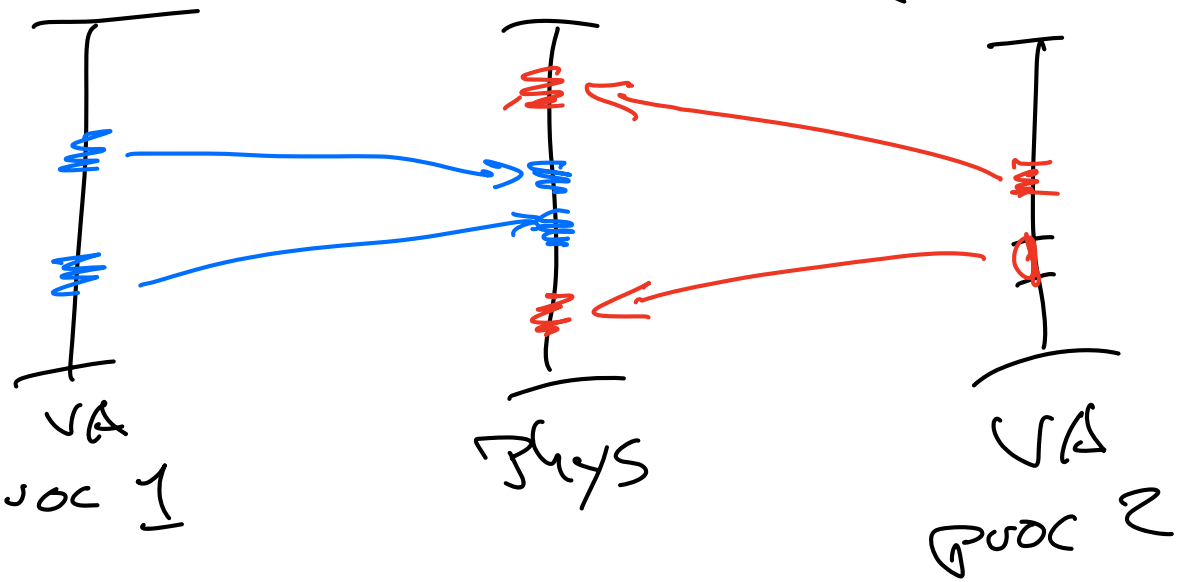
availability - no DOS denial of service

confidentiality: data ← ability to protect data from access
operations

integrity: data ← ability to protect from change (eg grade)
operations ← no trojans

How do computers provide security?

- lowest HW level - user/super mode



supervisor mode - can modify addr translation
user mode - can't " "
system calls, interrupts - specific known entry points

Possible scenarios:

- untrusted user binary code
  → the operating system problem

[- source code / Java etc bytecode]

- higher level - network services

actors

(users)

operations

read

write

:

add hook

clone

push

objects

file file2 file3 ...

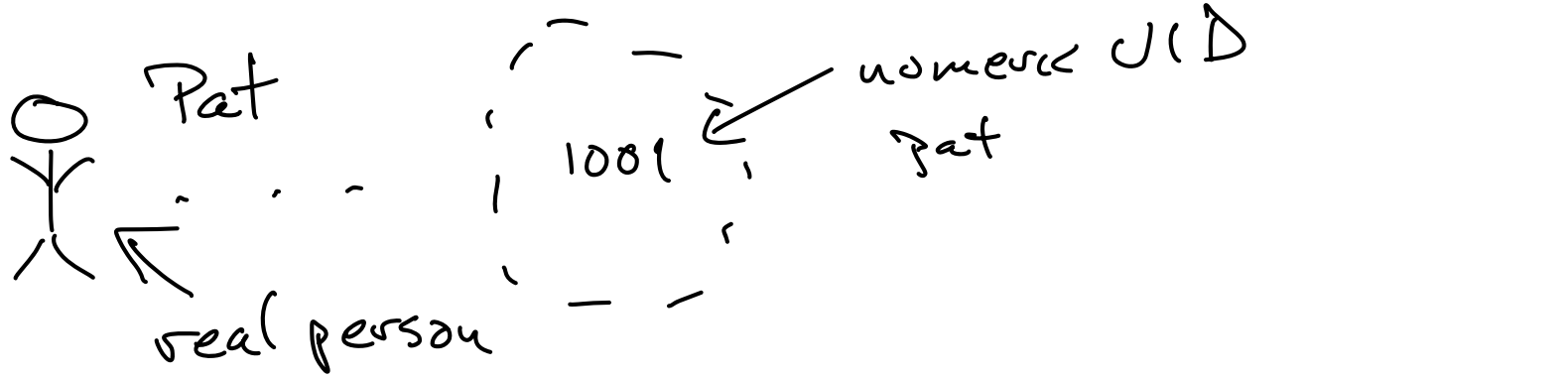
repo

directory

...

Actor - user.

Authentication



auth categories:

something you know : password

have : token, phone, etc

are : biometrics

(e.g. fingerprint)

Password authentication

user

① clear text password

② hashed passwd

```
main() {
  read pw
  check against /etc/passwd
  chuser(id)
  exec(/bin/sh)
}
```

password file

clear text

↓

user1 : passwd1
user2 : pw2
:

pw = read
hpw = hash(pw)
→ check against /etc/passwd

user1 : %1?2... (hash value)

dictionary attack

dict → hash ... per dictword

③ hash + "salt" (nonce)

pw + <rnd> → hash → ~~~
47

user1:47:~~~

verify pw:

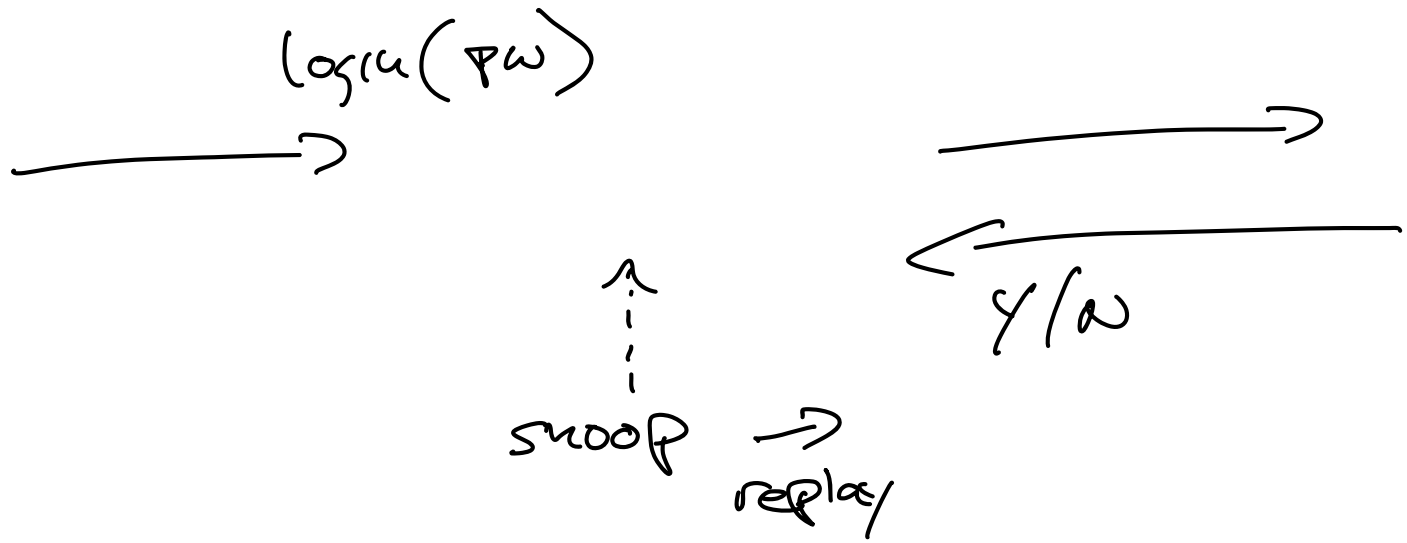
hash(pw+47) =? ~~~

↓

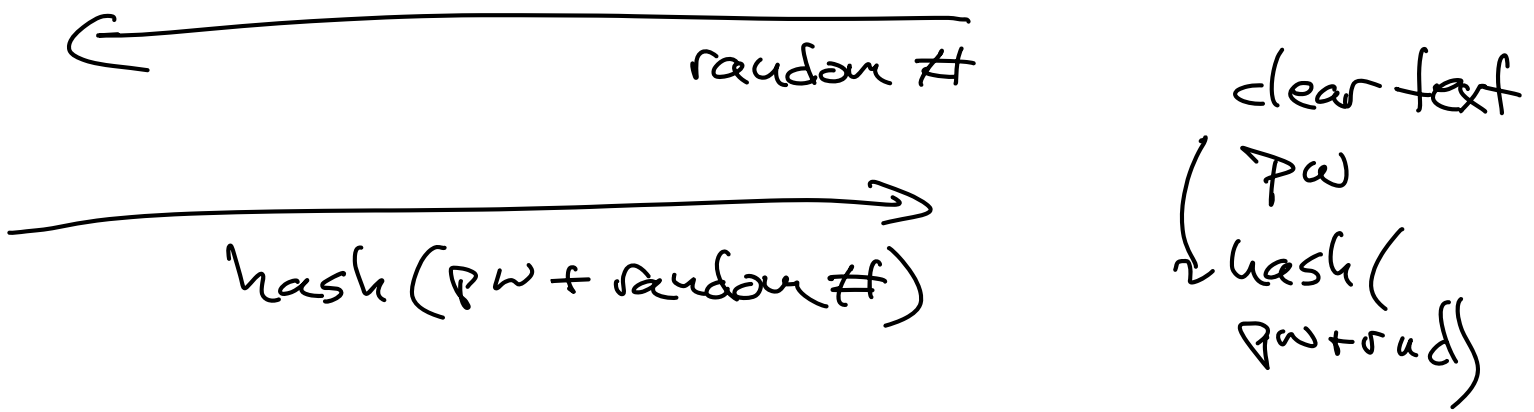
GPUs

④ hide the hash values (/etc/shadow)

The network password problem



① challenge-response



Public-key encryption

plain text → f(key1) → cyphertext

↳ f(key2) → plain text

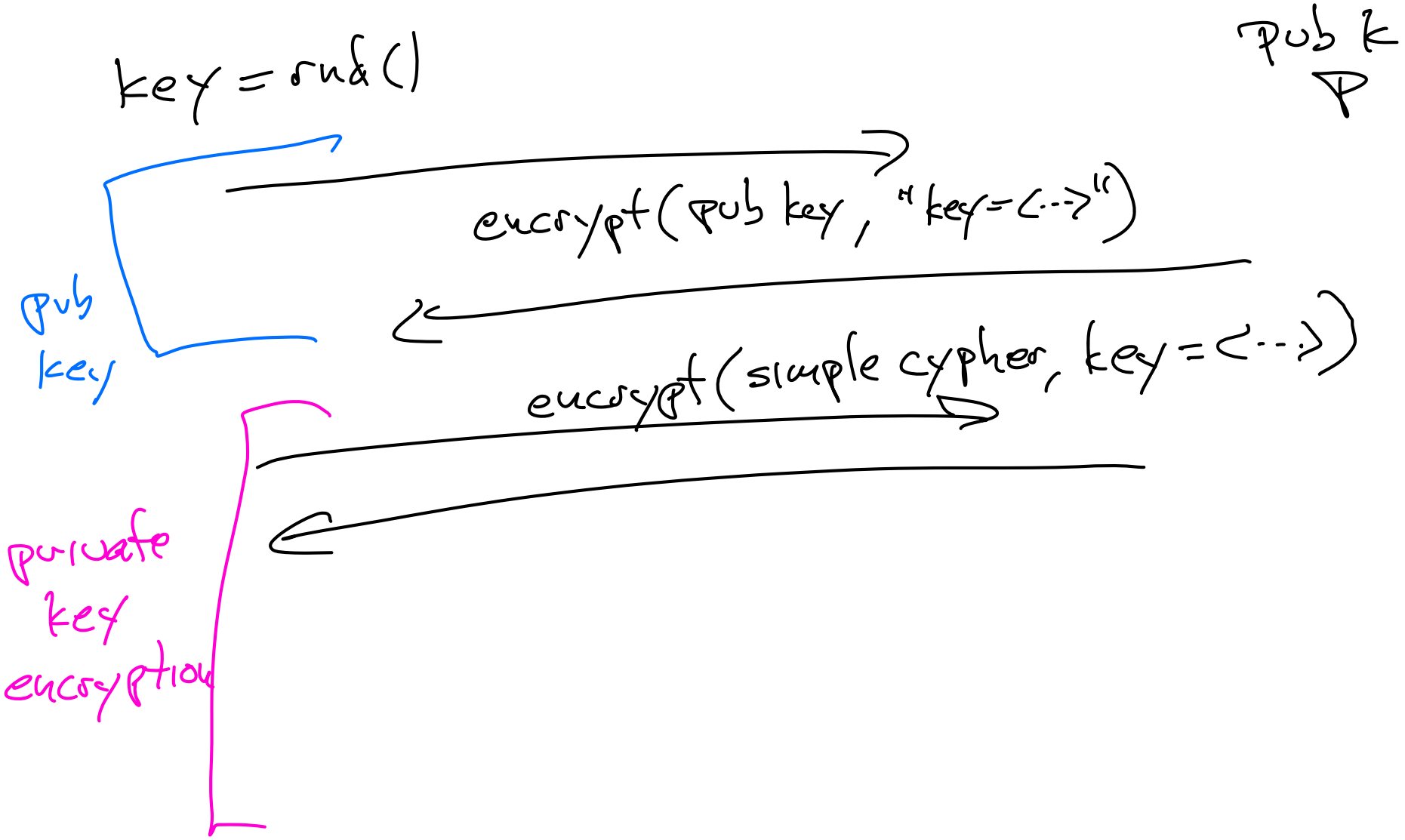
public key = prime1 * prime2

private = prime1, prime2

what can you do?

- send (-way messages (published pub. key)
- sign thruss

message → decrypt (priv) → encrypt → mss


key = rnd()
Pub k
P
encrypt(pub key, "key=<...>")
pub
key
encrypt(simple cypher, key=<...>)
private
key
encryption

Certificate signing

If we know pub key (A) verisign, etc

A provides:

B's public key is X
signed by A

joecoffeeco.com

priv public