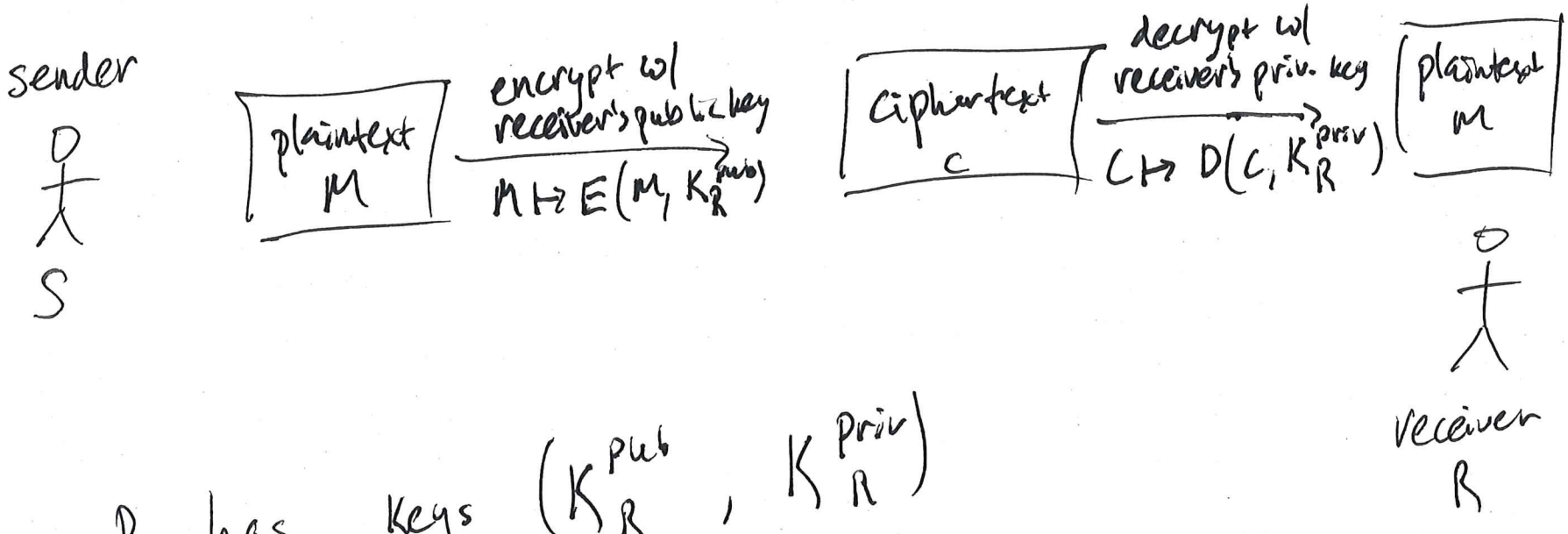
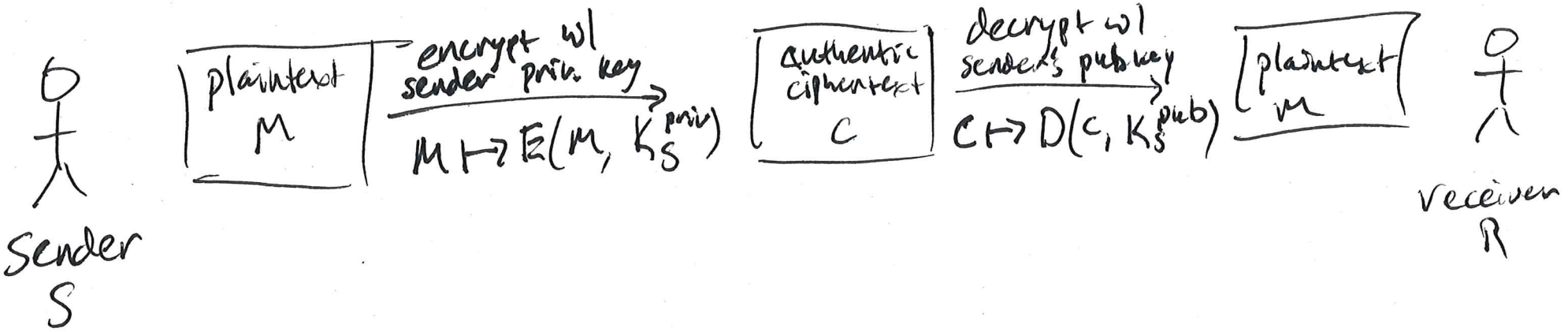
Public Key encryption



receiver R has keys (K_R^{pub}, K_R^{priv})

Public Key Signatures



Sender S has Keys (K_S^{pub}, K_S^{priv})

Lec 24

OS Security in Practice

Many Security bugs in OSes

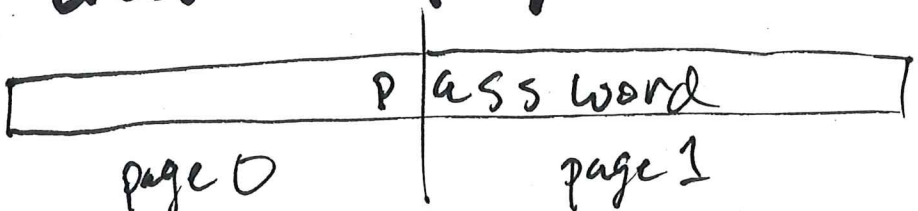
- attackers only need one way in
- Linux has hundreds of CVEs per year
- often attacks are unanticipated / creative
  - not just brute force password attack
- not necessarily technical — social engineering
- can't always tell that a breach occurred
  - can't easily recover from a breach

Tenex password attack

- Tenex was an early OS w/ demand paging
- code in kernel to check login password

```
check(entered, actual) {
    for (i=0; i< length; i++) {
        if (entered[i] != actual[i]) return false;
    }
    return true;
}
```

- user space can arrange for entered password to cross a page boundary



- arrange for page 0 to be resident but not page 1

Meltdown (2018)

- modern CPUs will speculatively execute instructions
  - eg:

```
%rax = X                    movq X, %rax
%rdi = *(base + %rax)       movq base(%rax), %rdi
```

since even L1 cache hit is slow
second instruction will be decoded + scheduled
even before first instruction retires

- if X is a secret, first instr will trap
  at which point second instruction should be squashed
  - except that base+x will still be in cache

Fixing Meltdown

- new CPUs do not modify cache until instruction is no longer speculative
- on old CPUs, can use different page tables in kernel / user mode
  - speculative execution will not have anything secret to read

Spectre (2018)

- like Meltdown but with mispredicted branch instead of privilege violation
- exploitable by any code on same core

Internet Worm (Morris '88)

Exploited 3 different problems to self replicate
- password dictionary
- sendmail debug interface
- fingerd buffer overflow

Would first check if target machine was already infected
but sometimes would randomly reinfect
↳ denial of service

Morris convicted of felony computer fraud + abuse

Ping of Death

- IP packets can be fragmented in flight
- reassembled at receiver by OS
  - buffer to hold fragments until all received
- each fragment contains an offset describing how to reassemble it
- early implementations did not check that offset + frag size < length of buffer

  ↳ buffer overflow
  - easy to cause receiver to crash
    possibly also remote code execution

Other worms

Code Red: exploited Microsoft IIS web server buffer overflow

- self replicating
- all infected machines would send requests to whitehouse.gov at same time

Ninda: multiple entry points
- made infected machines more vulnerable

Slammer: single packet exploit of MS SQL Server
- code just spammed this packet to random IPs overloading networks

Modern botnets

- botnet: pool of infected machines that can be controlled remotely
  - eg. to all simultaneously access same site
    DDOS
- there is a market for this

Unix talk

- users logged in to same machine could exchange messages in real time
- setuid root to write to other user's terminal
- bug in signal handler for Ctrl-C would drop you into a root shell
- TCB is not just OS kernel but also all setuid root programs

More Examples

- I/O DMA
- rowhammer
- embedded systems
  - cars
  - airplanes

24.10

Thompson's backdoor

Step 1: modify login.c

A:
```
if (username == "james") {
    don't check password;
    login as root;
}
```

Step 2: modify C compiler

B:
```
if (see trigger) {
    insert A into source program
}
```

and add trigger to login.C

Thompson's backdoor

Step 3: modify C compiler to hide step 2

```
C: if (see trigger2) {
       insert B into source program
       insert C into source program
   }
```

Then compile compiler with itself and save executable

Then replace C in source with trigger2