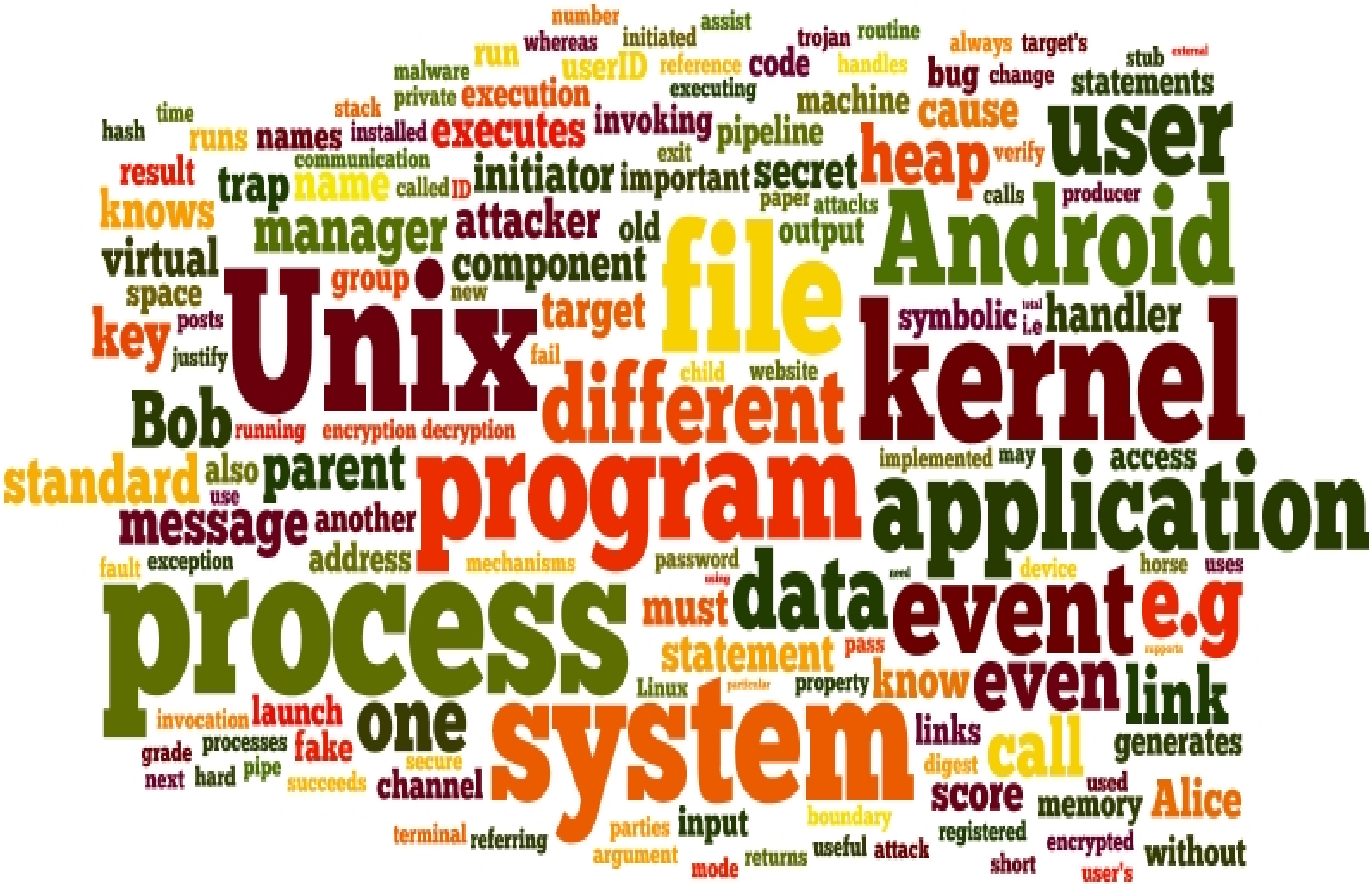
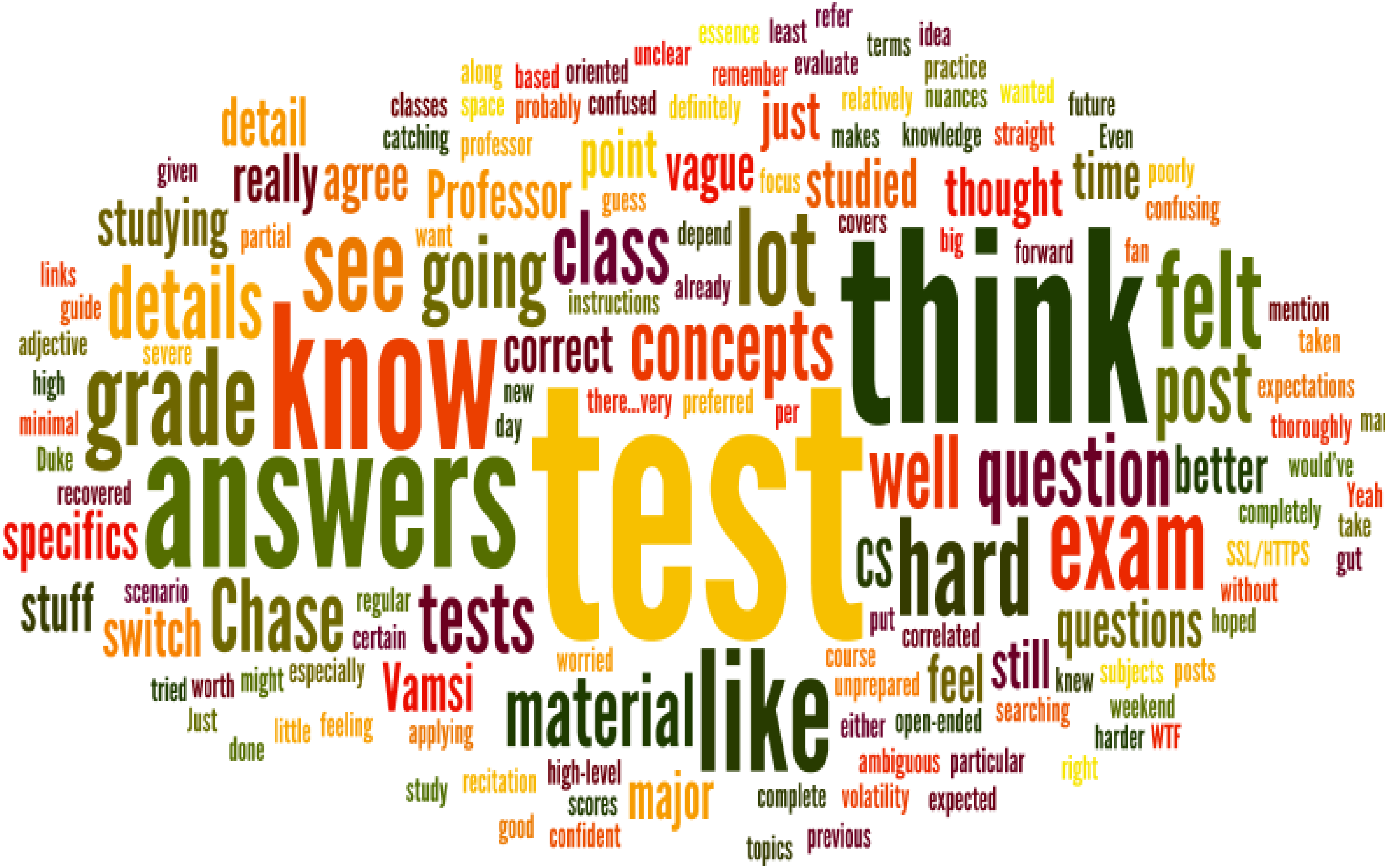
Midterm Exam Review

COMPSCI210 Recitation

5 Oct 2012

Vamsi Thummala


number
assist
whereas
initiated
trojan
routine
always
target's
external
run
userID
reference
code
handles
stub
malware
bug
change
executing
statements
private
execution
machine
stack
time
cause
hash
runs
names
installed
executes
invoking
pipeline
user
communication
exit
verify
result
heap
trap
name
initiator
important
secret
called
ID
calls
producer
paper
attacks
knows
manager
attacker
Android
old
output
virtual
component
file
group
space
new
symbolic
i.e
handler
Unix
target
key
posts
kernel
justify
fail
child
website
different
Bob
running
encryption
decryption
implemented
may
access
standard
also
parent
program
application
use
message
another
fault
exception
address
mechanisms
password
device
horse
uses
data
using
need
event
e.g
must
process
statement
pass
supports
Linux
particular
property
know
even
link
invocation
launch
one
system
links
grade
processes
fake
call
generates
secure
digest
next
hard
pipe
succeeds
channel
used
score
Alice
memory
boundary
input
terminal
referring
parties
registered
encrypted
argument
useful
attack
without
mode
returns
short
user's


essence
least
refer
idea
terms
unclear
evaluate
along
based
oriented
remember
practice
wanted
classes
space
probably
confused
definitely
relatively
nuances
future
detail
just
catching
professor
makes
knowledge
straight
Even
given
really
agree
Professor
point
vague
focus
studied
thought
time
poorly
guess
confusing
studying
partial
want
class
depend
lot
covers
big
forward
fan
links
see
going
think
felt
guide
details
already
instructions
mention
adjective
severe
correct
concepts
taken
know
post
high
expectations
minimal
grade
new
there...very
preferred
per
thoroughly
man
day
Duke
test
would've
well
question
better
recovered
Yeah
completely
specifics
answers
cs
exam
take
hard
SSL/HTTPS
gut
stuff
scenario
without
Chase
regular
tests
switch
put
certain
correlated
questions
hoped
worried
course
tried
worth
might
especially
Vamsi
unprepared
feel
still
knew
subjects
posts
weekend
Just
searching
little
feeling
applying
material
like
either
open-ended
done
harder
WTF
ambiguous
particular
right
high-level
study
recitation
major
complete
volatility
expected
scores
good
confident
topics
previous

My biased view

- A bit challenging
  - Tests your reading comprehension skills under time pressure
- Thought provoking
  - Not a brain dump test
- Opportunity for learning
  - Just an another class
- Tests your confidence
  - Not difficult to score but easy to get lost

Terminology so far ..

- Entity
  - Principal, Identity, Attribute, Label
- Component
- Context
- Connector
- Channel
- Trusted computing base
- Authentication
- Authorization
- Reference monitor
  - Subject, Object, Guard

- Process
- Thread
- Kernel
- Address space
- Files
- Pipes
- Sockets
- Binder
- Event

Systems/Abstractions

- Traditional single node
  - Unix
  - Linux
- Mobile
  - Android/Linux
- Web
  - Chromium/Linux
- Network basics
  - Client/Server

Security, an overview

We reduce it to three intertwined issues:

1. **What program am I running?**
   - Can this program be trusted? Who says?
   - Can I be sure that the program has not been tampered?
2. **Who am I talking to?**
   - Can this entity be trusted?
   - Can I be sure the communication has not been tampered?
3. **Should I approve this request?** R(op, subject, object)
   - Who is the requester? (subject)
   - What program is speaking for the requester?
   - Does the subject have the required permissions?

Elements of security

- **Isolation/protection**
  - Sandboxes and boundaries prevent unchecked access.
- **Integrity**
  - Fingerprint data to detect tampering.
  - Encrypt data to prevent access or tampering.
- **Authentication**
  - Identify a peer by proof that it possesses a secret.
- **Identity and attributes**
  - Identities have credentials: names, tags, roles...
- **Authorization == access control**
  - Guard checks credentials against an access policy.

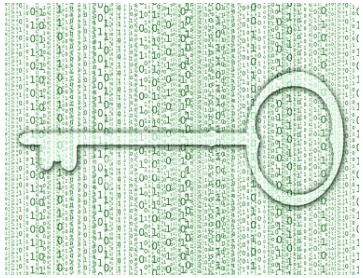




Access Control

- Triplet
  - {op, subject, object}
- Components run within contexts (isolated sandboxes).
- Each component/context is associated with an identity with some attributes (subject).
- Components use system calls to interact across context boundaries, or access shared objects.
- Each object has some access attributes.
- Principle of least privilege limits the damage a component can do if it “goes rogue”.

Crypto primitives



Encrypt/Decrypt

Signing

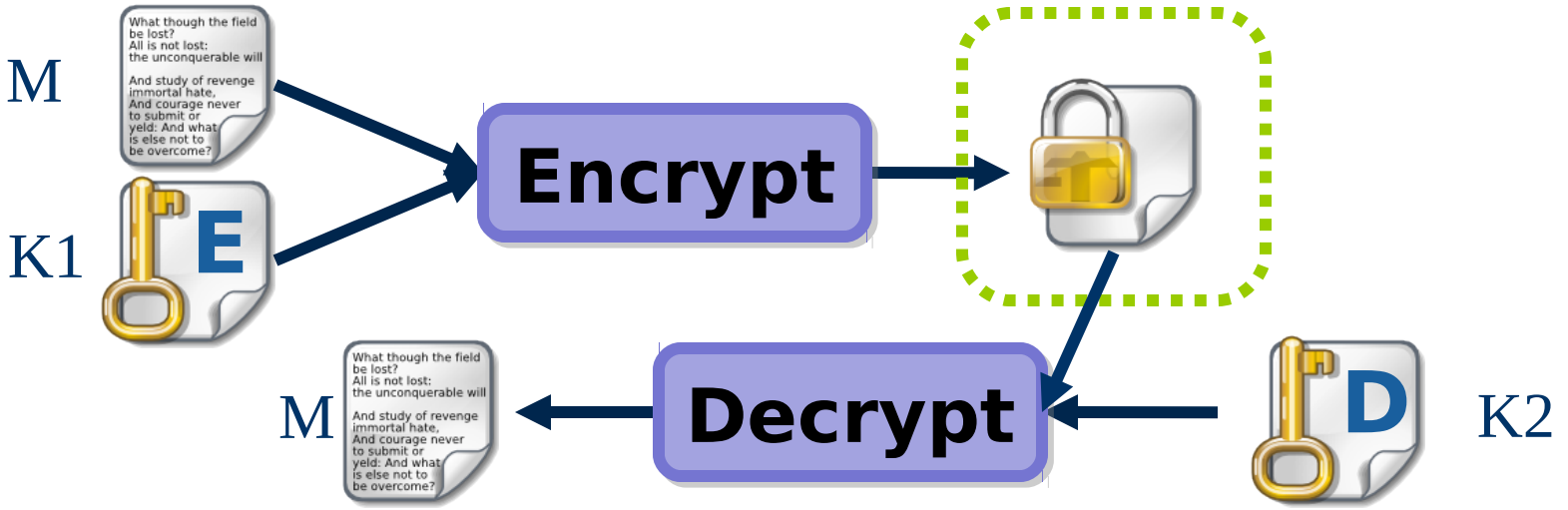
Use a shared secret key (symmetric) or use a keypair one public, one private (asymmetric)

Secure hashing

useful for fingerprinting data

Cryptography for Busy People

- **Standard crypto functions parameterized by keys.**
  - **Fixed-width "random" value (length matters, e.g., 256-bit)**
  - **Symmetric (DES: fast, requires shared key K1 = K2)**
  - **Asymmetric (RSA: slow, uses two keys)**
- **"Believed to be computationally infeasible" to break**



[Image: Landon Cox]

Two Flavors of “Signature”

- A digest encrypted with a private asymmetric key is called a digital signature
  - “Proves” that a particular identity sent the message.
    - “Proves” the message has not been tampered.
    - “Unforgeable”
  - The sender cannot deny sending the message.
    - “non-repudiable”
  - Can be legally binding in the United States
- A digest encrypted with a shared symmetric key is called a message authentication code (MAC).
    - faster, but…

Nonce

- Verifies the freshness of a message
- Eavesdropping
  - serverNonce
- Tampering
  - clientNonce