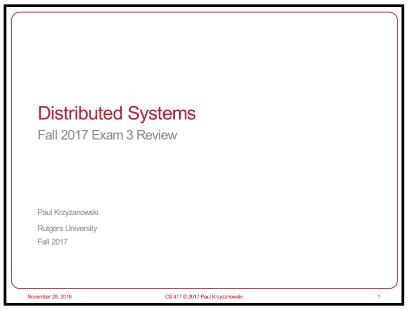


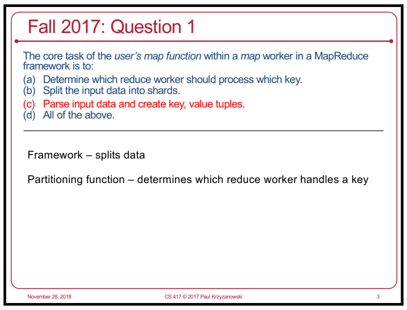
Distributed Systems

Fall 2017 Exam 3 Review

Paul Krzyzanowski

Rutgers University

Fall 2017

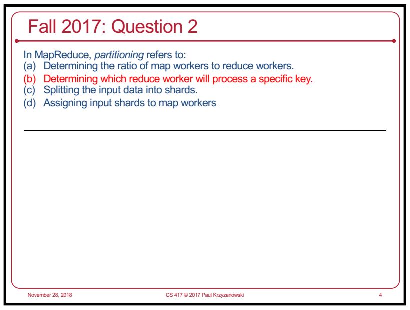


Fall 2017: Question 1

The core task of the *user's map function* within a *map* worker in a MapReduce framework is to:
(a) Determine which reduce worker should process which key.
(b) Split the input data into shards.
(c) Parse input data and create key, value tuples.
(d) All of the above.

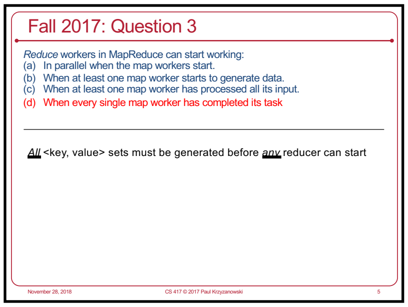
Framework – splits data

Partitioning function – determines which reduce worker handles a key



Fall 2017: Question 2

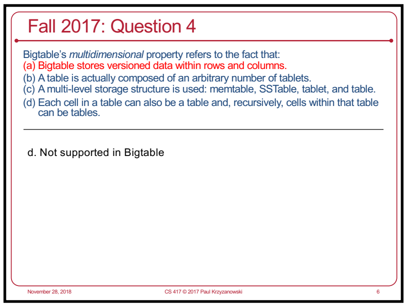
In MapReduce, *partitioning* refers to:
(a) Determining the ratio of map workers to reduce workers.
(b) Determining which reduce worker will process a specific key.
(c) Splitting the input data into shards.
(d) Assigning input shards to map workers



Fall 2017: Question 3

Reduce workers in MapReduce can start working:
(a) In parallel when the map workers start.
(b) When at least one map worker starts to generate data.
(c) When at least one map worker has processed all its input.
(d) When every single map worker has completed its task

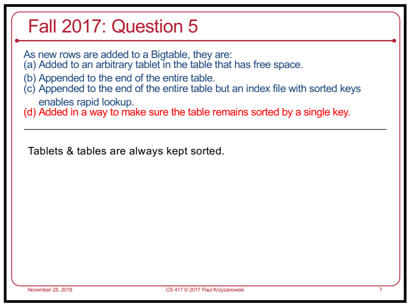
All <key, value> sets must be generated before *any* reducer can start



Fall 2017: Question 4

Bigtable's *multidimensional* property refers to the fact that:
(a) Bigtable stores versioned data within rows and columns.
(b) A table is actually composed of an arbitrary number of tablets.
(c) A multi-level storage structure is used: memtable, SSTable, tablet, and table.
(d) Each cell in a table can also be a table and, recursively, cells within that table can be tables.

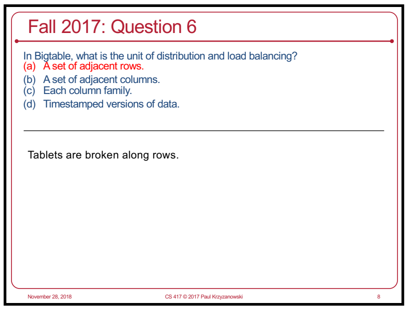
d. Not supported in Bigtable



Fall 2017: Question 5

As new rows are added to a Bigtable, they are:
(a) Added to an arbitrary tablet in the table that has free space.
(b) Appended to the end of the entire table.
(c) Appended to the end of the entire table but an index file with sorted keys enables rapid lookup.
(d) Added in a way to make sure the table remains sorted by a single key.

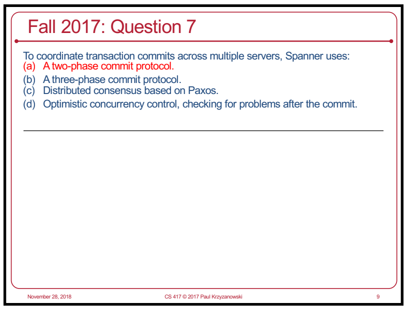
Tablets & tables are always kept sorted.



Fall 2017: Question 6

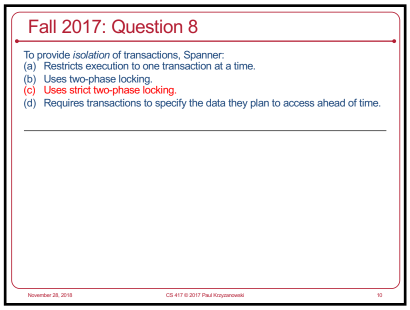
In Bigtable, what is the unit of distribution and load balancing?
(a) A set of adjacent rows.
(b) A set of adjacent columns.
(c) Each column family.
(d) Timestamped versions of data.

Tablets are broken along rows.



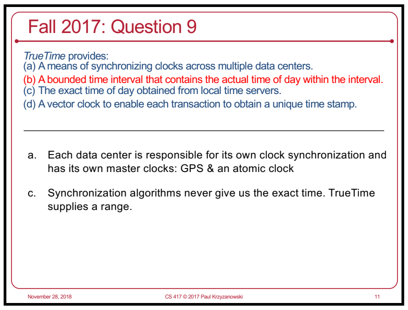
Fall 2017: Question 7

To coordinate transaction commits across multiple servers, Spanner uses:
(a) A two-phase commit protocol.
(b) A three-phase commit protocol.
(c) Distributed consensus based on Paxos.
(d) Optimistic concurrency control, checking for problems after the commit.



Fall 2017: Question 8

To provide *isolation* of transactions, Spanner:
(a) Restricts execution to one transaction at a time.
(b) Uses two-phase locking.
(c) Uses strict two-phase locking.
(d) Requires transactions to specify the data they plan to access ahead of time.

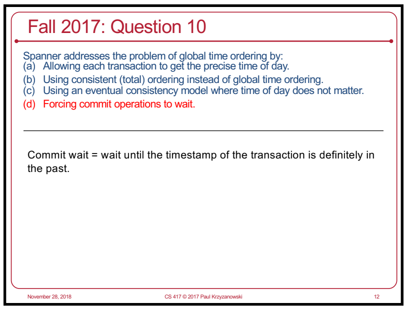


Fall 2017: Question 9

TrueTime provides:
(a) A means of synchronizing clocks across multiple data centers.
(b) A bounded time interval that contains the actual time of day within the interval.
(c) The exact time of day obtained from local time servers.
(d) A vector clock to enable each transaction to obtain a unique time stamp.

a. Each data center is responsible for its own clock synchronization and has its own master clocks: GPS & an atomic clock

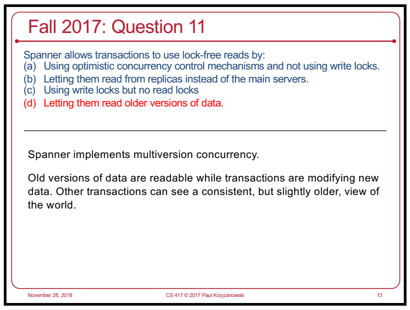
c. Synchronization algorithms never give us the exact time. TrueTime supplies a range.



Fall 2017: Question 10

Spanner addresses the problem of global time ordering by:
(a) Allowing each transaction to get the precise time of day.
(b) Using consistent (total) ordering instead of global time ordering.
(c) Using an eventual consistency model where time of day does not matter.
(d) Forcing commit operations to wait.

Commit wait = wait until the timestamp of the transaction is definitely in the past.

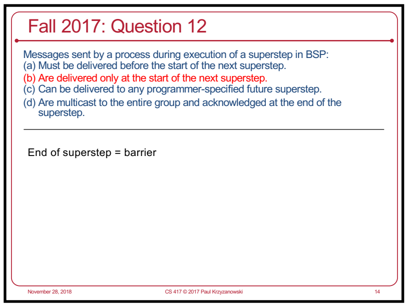


Fall 2017: Question 11

Spanner allows transactions to use lock-free reads by:
(a) Using optimistic concurrency control mechanisms and not using write locks.
(b) Letting them read from replicas instead of the main servers.
(c) Using write locks but no read locks
(d) Letting them read older versions of data.

Spanner implements multiversion concurrency.

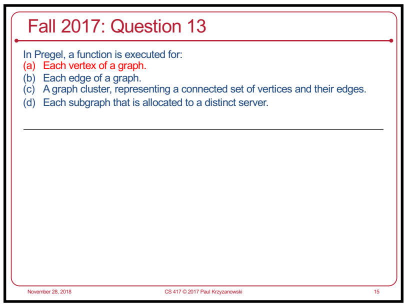
Old versions of data are readable while transactions are modifying new data. Other transactions can see a consistent, but slightly older, view of the world.



Fall 2017: Question 12

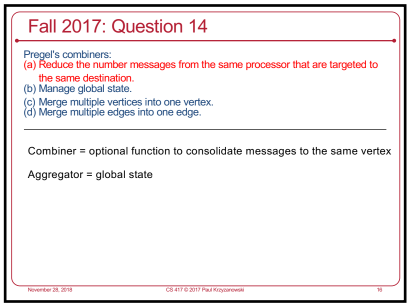
Messages sent by a process during execution of a superstep in BSP:
(a) Must be delivered before the start of the next superstep.
(b) Are delivered only at the start of the next superstep.
(c) Can be delivered to any programmer-specified future superstep.
(d) Are multicast to the entire group and acknowledged at the end of the superstep.

End of superstep = barrier



Fall 2017: Question 13

In Pregel, a function is executed for:
(a) Each vertex of a graph.
(b) Each edge of a graph.
(c) A graph cluster, representing a connected set of vertices and their edges.
(d) Each subgraph that is allocated to a distinct server.

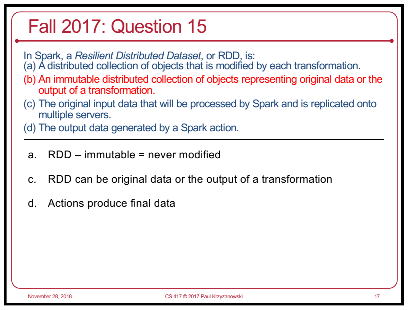


Fall 2017: Question 14

Pregel's combiners:
(a) Reduce the number messages from the same processor that are targeted to the same destination.
(b) Manage global state.
(c) Merge multiple vertices into one vertex.
(d) Merge multiple edges into one edge.

Combiner = optional function to consolidate messages to the same vertex

Aggregator = global state



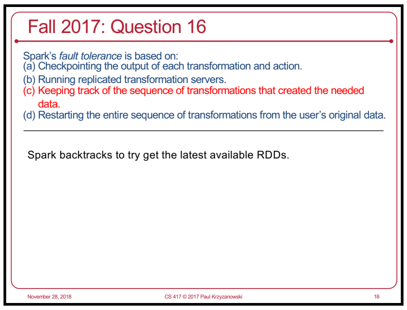
Fall 2017: Question 15

In Spark, a *Resilient Distributed Dataset*, or RDD, is:
(a) A distributed collection of objects that is modified by each transformation.
(b) An immutable distributed collection of objects representing original data or the output of a transformation.
(c) The original input data that will be processed by Spark and is replicated onto multiple servers.
(d) The output data generated by a Spark action.

a. RDD – immutable = never modified

c. RDD can be original data or the output of a transformation

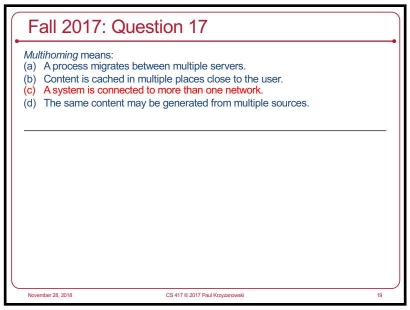
d. Actions produce final data



Fall 2017: Question 16

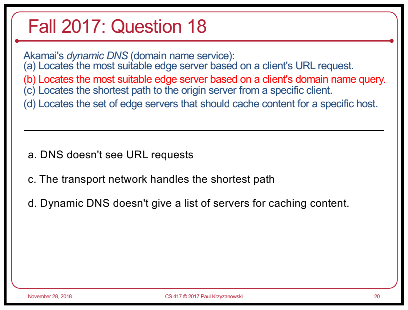
Spark's *fault tolerance* is based on:
(a) Checkpointing the output of each transformation and action.
(b) Running replicated transformation servers.
(c) Keeping track of the sequence of transformations that created the needed data.
(d) Restarting the entire sequence of transformations from the user's original data.

Spark backtracks to try get the latest available RDDs.



Fall 2017: Question 17

Multihoming means:
(a) A process migrates between multiple servers.
(b) Content is cached in multiple places close to the user.
(c) A system is connected to more than one network.
(d) The same content may be generated from multiple sources.



Fall 2017: Question 18

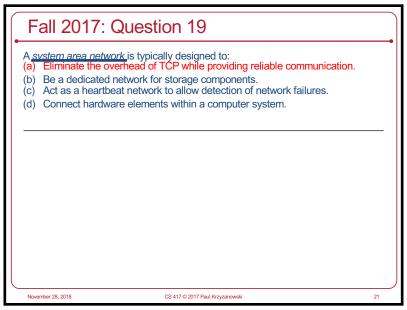
Akamai's *dynamic DNS* (domain name service):
(a) Locates the most suitable edge server based on a client's URL request.
(b) Locates the most suitable edge server based on a client's domain name query.
(c) Locates the shortest path to the origin server from a specific client.
(d) Locates the set of edge servers that should cache content for a specific host.

a. DNS doesn't see URL requests

c. The transport network handles the shortest path

d. Dynamic DNS doesn't give a list of servers for caching content.

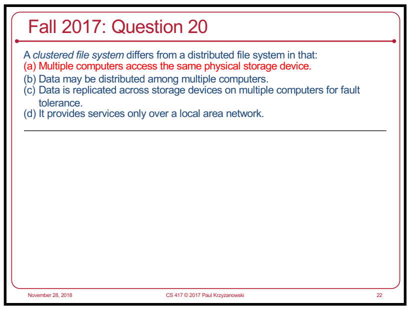
November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 20



Fall 2017: Question 19

A *system area network* is typically designed to:
(a) Eliminate the overhead of TCP while providing reliable communication.
(b) Be a dedicated network for storage components.
(c) Act as a heartbeat network to allow detection of network failures.
(d) Connect hardware elements within a computer system.

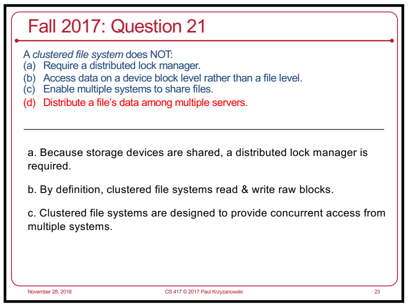
November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 21



Fall 2017: Question 20

A *clustered file system* differs from a distributed file system in that:
(a) Multiple computers access the same physical storage device.
(b) Data may be distributed among multiple computers.
(c) Data is replicated across storage devices on multiple computers for fault tolerance.
(d) It provides services only over a local area network.

November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 22



Fall 2017: Question 21

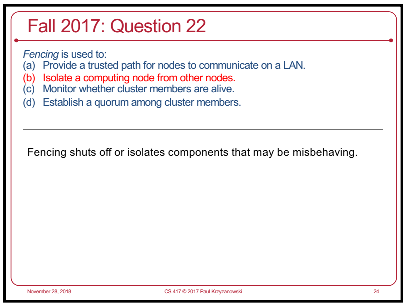
A *clustered file system* does NOT:
(a) Require a distributed lock manager.
(b) Access data on a device block level rather than a file level.
(c) Enable multiple systems to share files.
(d) Distribute a file's data among multiple servers.

a. Because storage devices are shared, a distributed lock manager is required.

b. By definition, clustered file systems read & write raw blocks.

c. Clustered file systems are designed to provide concurrent access from multiple systems.

November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 23

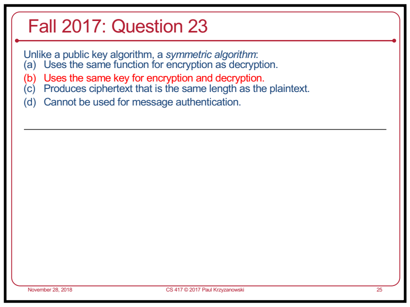


Fall 2017: Question 22

Fencing is used to:
(a) Provide a trusted path for nodes to communicate on a LAN.
(b) Isolate a computing node from other nodes.
(c) Monitor whether cluster members are alive.
(d) Establish a quorum among cluster members.

Fencing shuts off or isolates components that may be misbehaving.

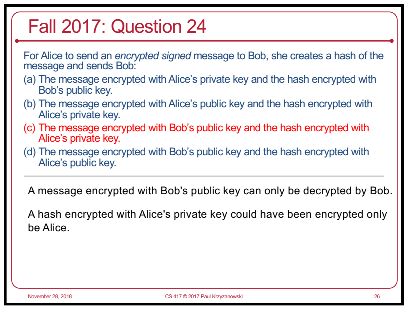
November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 24



Fall 2017: Question 23

Unlike a public key algorithm, a *symmetric algorithm*:
(a) Uses the same function for encryption as decryption.
(b) Uses the same key for encryption and decryption.
(c) Produces ciphertext that is the same length as the plaintext.
(d) Cannot be used for message authentication.

November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 25



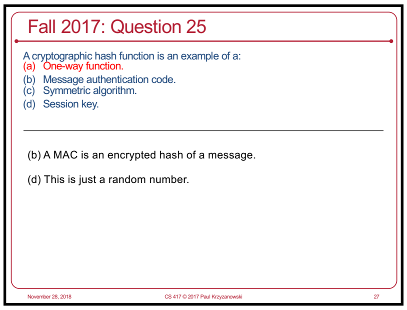
Fall 2017: Question 24

For Alice to send an *encrypted signed* message to Bob, she creates a hash of the message and sends Bob:
(a) The message encrypted with Alice's private key and the hash encrypted with Bob's public key.
(b) The message encrypted with Alice's public key and the hash encrypted with Alice's private key.
(c) The message encrypted with Bob's public key and the hash encrypted with Alice's private key.
(d) The message encrypted with Bob's public key and the hash encrypted with Alice's public key.

A message encrypted with Bob's public key can only be decrypted by Bob.

A hash encrypted with Alice's private key could have been encrypted only be Alice.

November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 26



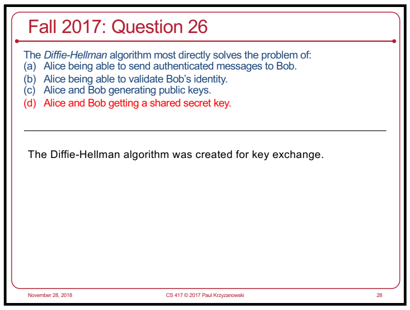
Fall 2017: Question 25

A cryptographic hash function is an example of a:
(a) One-way function.
(b) Message authentication code.
(c) Symmetric algorithm.
(d) Session key.

(b) A MAC is an encrypted hash of a message.

(d) This is just a random number.

November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 27

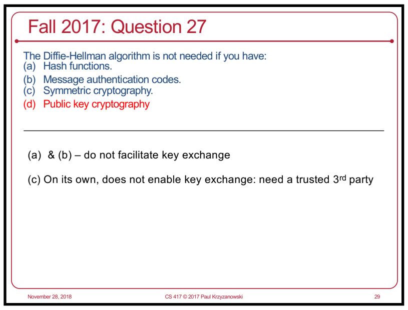


Fall 2017: Question 26

The *Diffie-Hellman* algorithm most directly solves the problem of:
(a) Alice being able to send authenticated messages to Bob.
(b) Alice being able to validate Bob's identity.
(c) Alice and Bob generating public keys.
(d) Alice and Bob getting a shared secret key.

The Diffie-Hellman algorithm was created for key exchange.

November 28, 2018 CS 417 © 2017 Paul Krzyzanowski 28

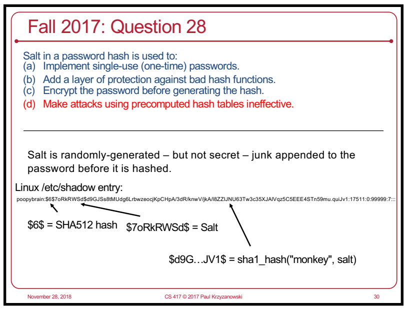


Fall 2017: Question 27

The Diffie-Hellman algorithm is not needed if you have:
(a) Hash functions.
(b) Message authentication codes.
(c) Symmetric cryptography.
(d) Public key cryptography

(a) & (b) – do not facilitate key exchange

(c) On its own, does not enable key exchange: need a trusted 3rd party



Fall 2017: Question 28

Salt in a password hash is used to:
(a) Implement single-use (one-time) passwords.
(b) Add a layer of protection against bad hash functions.
(c) Encrypt the password before generating the hash.
(d) Make attacks using precomputed hash tables ineffective.

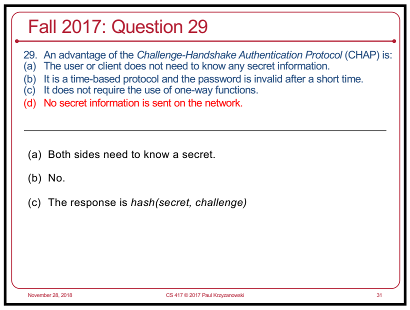
Salt is randomly-generated – but not secret – junk appended to the password before it is hashed.

Linux /etc/shadow entry:

```
poopybrain:$6$7oRkRWSd$d9GJSs8tMUdg6LrbwzeocjKpCHpA/3dR/knwV/jkA/I8ZZIJNU63Tw3c35XJAlVqz5C5EEE4STn59mu.quiJv1:17511:0:99999:7:::
```

$6$ = SHA512 hash $7oRkRWSd$ = Salt

$d9G…JV1$ = sha1_hash("monkey", salt)



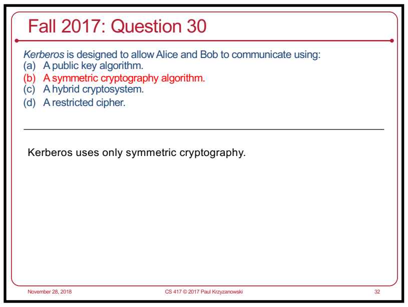
Fall 2017: Question 29

29. An advantage of the *Challenge-Handshake Authentication Protocol* (CHAP) is:
(a) The user or client does not need to know any secret information.
(b) It is a time-based protocol and the password is invalid after a short time.
(c) It does not require the use of one-way functions.
(d) No secret information is sent on the network.

(a) Both sides need to know a secret.

(b) No.

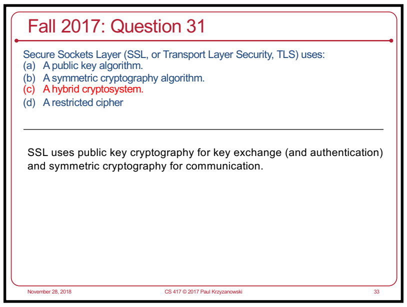
(c) The response is *hash(secret, challenge)*



Fall 2017: Question 30

Kerberos is designed to allow Alice and Bob to communicate using:
(a) A public key algorithm.
(b) A symmetric cryptography algorithm.
(c) A hybrid cryptosystem.
(d) A restricted cipher.

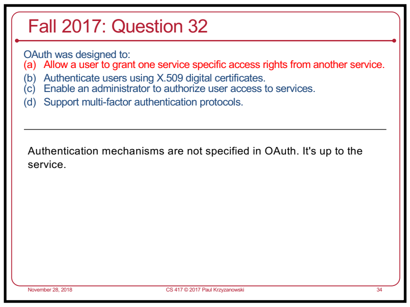
Kerberos uses only symmetric cryptography.



Fall 2017: Question 31

Secure Sockets Layer (SSL, or Transport Layer Security, TLS) uses:
(a) A public key algorithm.
(b) A symmetric cryptography algorithm.
(c) A hybrid cryptosystem.
(d) A restricted cipher

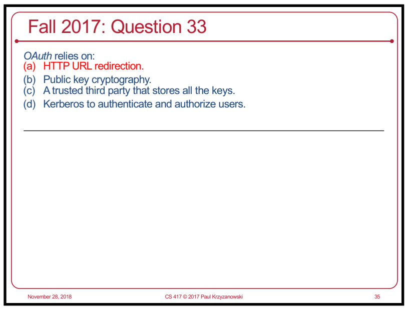
SSL uses public key cryptography for key exchange (and authentication) and symmetric cryptography for communication.



Fall 2017: Question 32

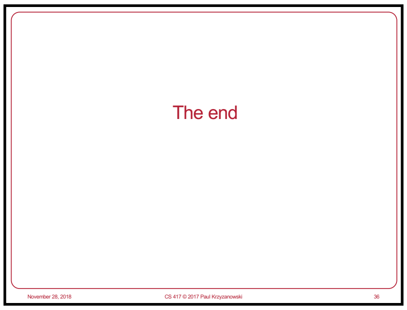
OAuth was designed to:
(a) Allow a user to grant one service specific access rights from another service.
(b) Authenticate users using X.509 digital certificates.
(c) Enable an administrator to authorize user access to services.
(d) Support multi-factor authentication protocols.

Authentication mechanisms are not specified in OAuth. It's up to the service.



Fall 2017: Question 33

OAuth relies on:
(a) HTTP URL redirection.
(b) Public key cryptography.
(c) A trusted third party that stores all the keys.
(d) Kerberos to authenticate and authorize users.



The end