CS162
Operating Systems and Systems Programming
Lecture 25

Security (cont), Cloud Services & Cloud Infrastructure

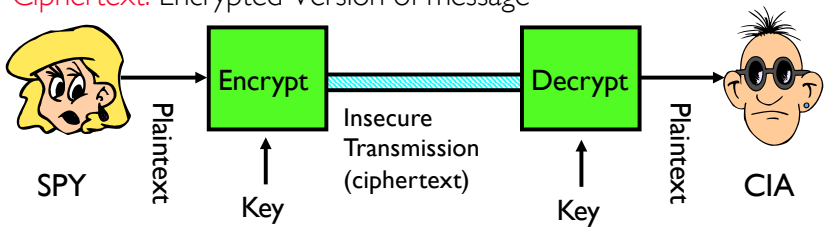
November 28th, 2016

Prof. Anthony D. Joseph

http://cs162.eecs.Berkeley.edu

Recall: Private Key Cryptography

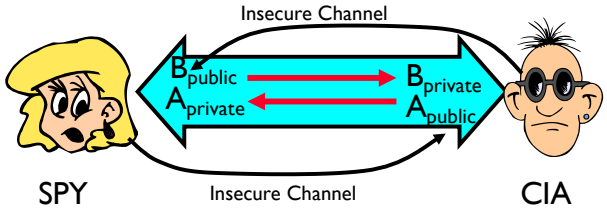
- Private Key (Symmetric) Encryption:
  - Single key used for both encryption and decryption
- Plaintext: Unencrypted Version of message
- Ciphertext: Encrypted Version of message



- Important properties
  - Can't derive plain text from ciphertext (decode) without access to key
  - Can't derive key from plain text and ciphertext
  - As long as password stays secret, get both secrecy and authentication
- Symmetric Key Algorithms: DES, Triple-DES, AES

Recall: Public Key Encryption Details

- Idea: K_{public} can be made public, keep $K_{private}$ private

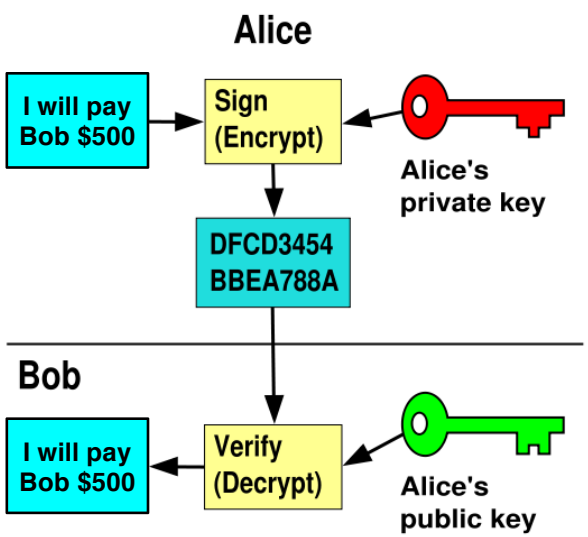


- Gives message privacy (restricted receiver):
  - Public keys (secure destination points) can be acquired by anyone/used by anyone
  - Only person with private key can decrypt message
- What about authentication?
  - Use combination of private and public key
  - Alice→Bob: $[(\text{I'm Alice})^{A_{private}}\ \text{Rest of message}]^{B_{public}}$
  - Provides restricted sender and receiver
- But: how does Alice know that it was Bob who sent her B_{public}? And vice versa…
  - Need a certificate authority to sign keys!

Non-Repudiation: RSA Crypto & Signatures

- Suppose Alice has published public key K_E
- If she wishes to prove who she is, she can send a message x encrypted with her private key K_D (i.e., she sends $E(x, K_D)$)
  - Anyone knowing Alice's public key K_E can recover x, verify that Alice must have sent the message
    - » It provides a signature
  - Alice can't deny it ⇒ non-repudiation
- Could simply encrypt a hash of the data to sign a document that you wanted to be in clear text
- Note that either of these signature techniques work perfectly well with any data (not just messages)
  - Could sign every datum in a database, for instance

RSA Crypto & Signatures (cont'd)



Digital Certificates

- How do you know K_E is Alice's public key?
- Trusted authority (e.g., Verisign) signs binding between Alice and K_E with its private key $KV_{private}$
  - $C = E(\{Alice, K_E\}, KV_{private})$
  - C: digital certificate
- Alice: distribute her digital certificate, C
- Anyone: use trusted authority's KV_{public}, to extract Alice's public key from C
  - $D(C, KV_{public}) =$
    $D(E(\{Alice, K_E\}, KV_{private}), KV_{public}) = \{Alice, K_E\}$

Summary of Our Crypto Toolkit

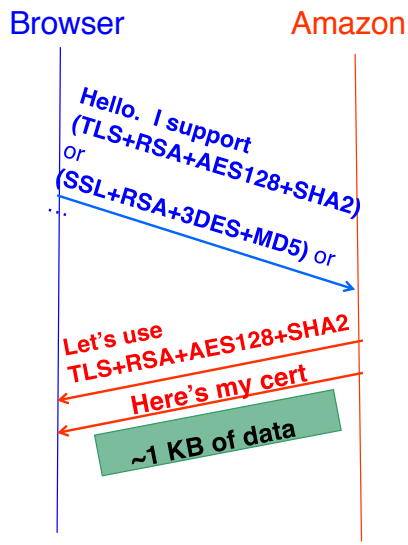
- If we can securely distribute a key, then
  - Symmetric ciphers (e.g., AES) offer fast, presumably strong confidentiality
- Public key cryptography does away with (potentially major) problem of secure key distribution
  - But: not as computationally efficient
    - » Often addressed by using public key crypto to exchange a session key
- Digital signature binds the public key to an entity

Putting It All Together - HTTPS

- What happens when you click on https://www.amazon.com?
- `https` = "Use HTTP over SSL/TLS"
  - SSL = Secure Socket Layer
  - TLS = Transport Layer Security
    - » Successor to SSL
  - Provides security layer (authentication, encryption) on top of TCP
    - » Fairly transparent to applications

HTTPS Connection (SSL/TLS) (cont'd)

- Browser (client) connects via TCP to Amazon's HTTPS server
- Client sends over list of crypto protocols it supports
- Server picks protocols to use for this session
- Server sends over its certificate
- (all of this is in the clear)



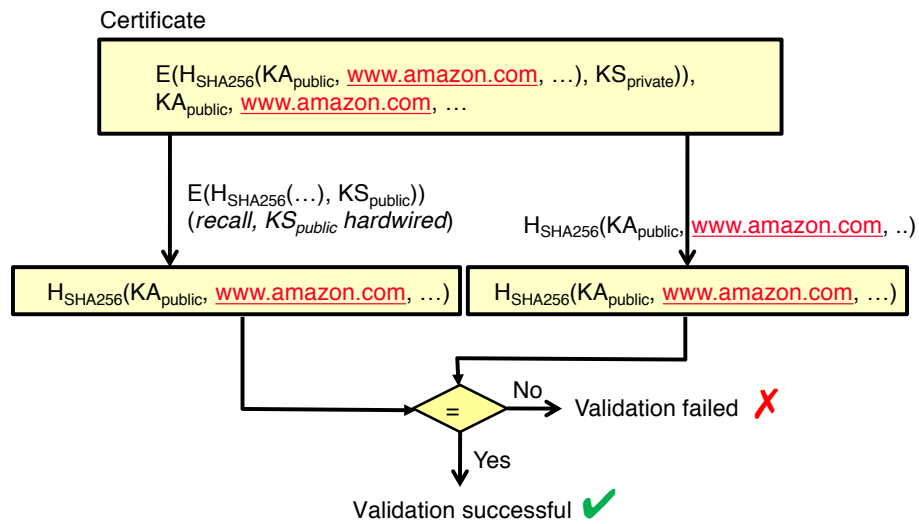
Inside the Server's Certificate

- Name associated with cert (e.g., Amazon)
- Amazon's RSA public key
- A bunch of auxiliary info (physical address, type of cert, expiration time)
- Name of certificate's signatory (who signed it)
- A public-key signature of a hash (SHA-256) of all this
  - Constructed using the signatory's private RSA key, i.e.,
  - Cert = $E(H_{SHA256}(KA_{public}, \text{www.amazon.com}, \ldots), KS_{private})$)
    - » KA_{public}: Amazon's public key
    - » $KS_{private}$: signatory (certificate authority) private key
- …

Validating Amazon's Identity

- How does the browser authenticate certificate signatory?
  - Certificates of several certificate authorities (e.g., Verisign) are hardwired into the browser (or OS)
- If can't find cert, warn user that site has not been verified
  - And may ask whether to continue
  - Note, can still proceed, just without authentication
- Browser uses public key in signatory's cert to decrypt signature
  - Compares with its own SHA-256 hash of Amazon's cert
- Assuming signature matches, now have high confidence it's indeed Amazon … assuming signatory is trustworthy
  - DigiNotar CA breach (July-Sept 2011): Google, Yahoo!, Mozilla, Tor project, Wordpress, … (531 total certificates)

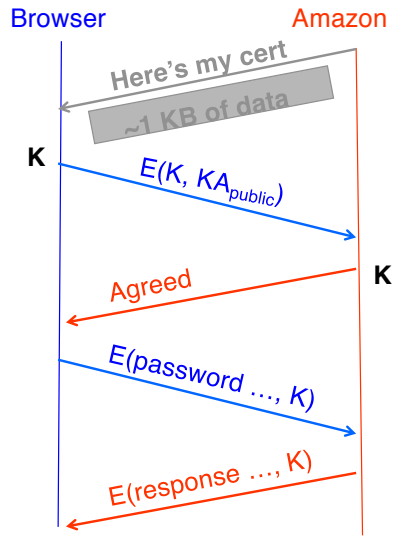
Certificate Validation



Can also validate using peer approach: https://www.eff.org/observatory

HTTPS Connection (SSL/TLS) cont'd

- Browser constructs a random session key K used for data communication
  - Private key for bulk crypto
- Browser encrypts K using Amazon's public key
- Browser sends E(K, KA_{public}) to server
- Browser displays 🔒
- All subsequent comm. encrypted w/ symmetric cipher (e.g., AES128) using key K
  - E.g., client can authenticate using a password



Security Summary

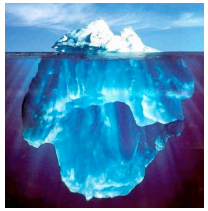
- Many more challenges to building secure systems and applications
- No fixed-point solutions
- Adversaries constantly change and adapt
- Defenses must also constantly change and adapt
- Take CS 161

Administrivia

- Prof Joseph's Tue office hours rescheduled to Wed Noon-1pm in 465F Soda
- Midterm #3 on **Wednesday 11/30 5-6:30PM**
  - 1 LeConte (Last name A-H) and 2050 VLSB (Last name I-Z)
  - Topics primarily course material from lectures 16 – 25
    - » Lectures, projects, homeworks, readings, textbook
  - Closed book, no calculators, ***three* double-side letter-sized page of handwritten notes**
  - Review after class on today Monday 11/28 in 2050 VLSB
- Project #3 code due next Monday 12/2

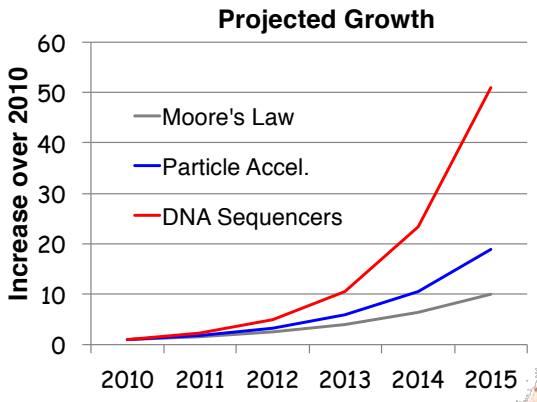
Data Deluge

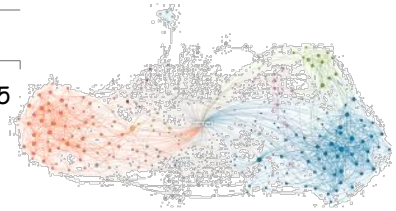
- Billions of users connected through the net
  - WWW, FB, twitter, cell phones, …
  - 80% of the data on FB was produced last year
  - FB building Exabyte ($2^{60} \cong 10^{18}$) data centers



- It's all happening online – could record every:
  - Click, ad impression, billing event, server request, transaction, network msg, fault, fast forward, pause, skip, …
- User Generated Content (Web & Mobile)
  - Facebook, Instagram, Yelp, TripAdvisor, Twitter, YouTube, …

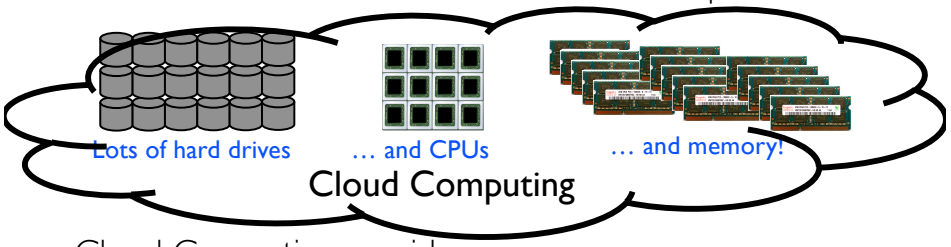
Data Grows Faster than Moore's Law





The Big Data Solution: Cloud Computing

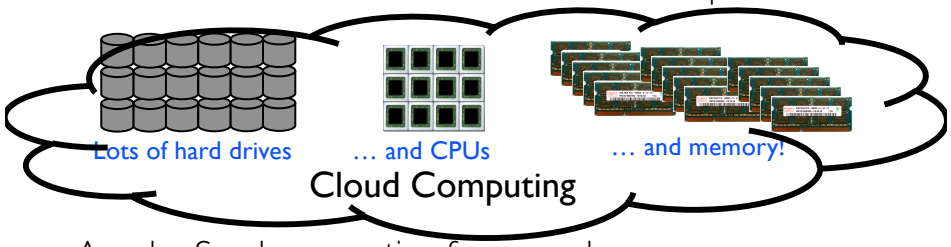
- One machine can not process or even store all the data!
- Solution: distribute data over cluster of cheap machines



- Cloud Computing provides:
  - Illusion of infinite resources
  - Short-term, on-demand resource allocation
  - Can be much less expensive than owning computers
  - Access to latest technologies (SSDs, GPUs, …)

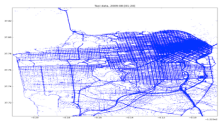
The Big Data Solution: Cloud Computing

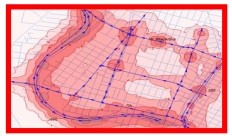
- One machine can not process or even store all the data!
- Solution: distribute data over cluster of cheap machines



- Apache Spark computing framework:
  - Programming abstraction and parallel runtime to hide complexities of fault-tolerance and slow machines
  - *"Here's an operation, run it on all of the data"*
  - I don't care where it runs (you schedule that)
  - In fact, feel free to run it twice on different nodes

What Can You do with Big Data?





Crowdsourcing + Physical modeling + Sensing + Data Assimilation

=

http://traffic.berkeley.edu

Real World Spark and Cloud Computing Use Cases

- My research (today):
  - Secure Machine Learning
  - Big Data Genomics using ADAM
- Other use cases:
  - Conviva optimizing Internet video stream delivery
  - Data processing for wearables and Internet of Things
  - Personalized Yahoo! news pages
  - Analytics for Yahoo! advertising
  - Capital One product recommendations

Secure Machine Learning: Our View

- Security Analytics: Using Robust ML for adversary resistant security metrics and analytics
  - *Pattern mining and prediction, at scale, on big data, with adversaries*
  - Detecting and classifying malware, spam, and malicious sites/URLs
  - Identifying authors of User Generated Content and malware
- Situational Awareness: Helping the humans-in-the-loop
  - *Real-time, Machine Learning-based analytics for humans*
  - Domain experts
  - Crowds and human reviewers
  - End-users of systems

Adversarial Exploitation of ML

- Traditional approach – *Evading Adversary*
  - Attacker determines decision boundary
  - Crafts (positive instance) content that is classified as negative
- Input data "drifts" over time, so must periodically retrain ML
  - Use previously classified input data
- Newer approach – *Influencing Adversary*
  - *Patient* attacker operates during periodic retraining stage by injecting "tricky" positive instances
  - Shifts decision boundary over time during retraining such that (positive instance) content is eventually classified as negative
- Need novel adaptive, robust ML techniques to defend against Influencing Adversaries – already a threat for spam detection

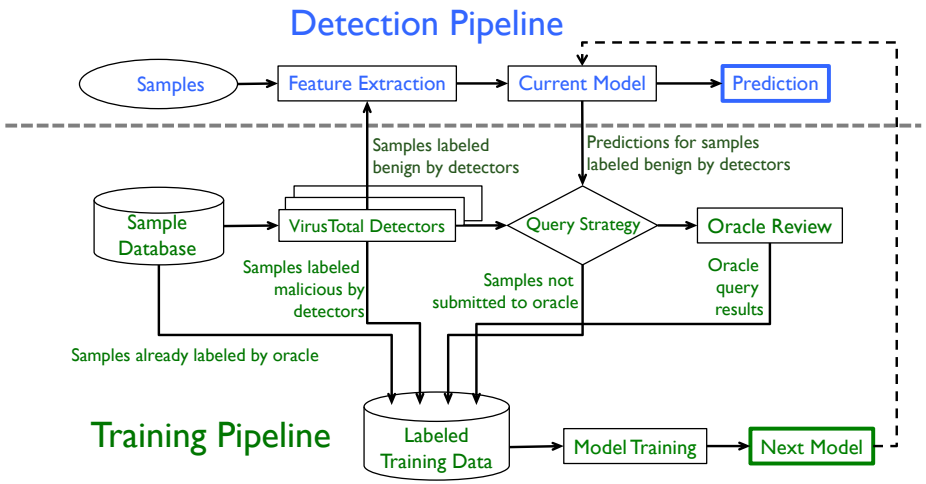
An ML Analysis Pipeline: From Labs to Production

- Goal: research AND production environment
  - ***Moonshot: Detect undetectable Advanced Persistent Threats***
- Significant challenges to deploying ML in practice:
  - Increased human insight – *how the system is making decisions*
  - Scalable, interactive analysis – *large-scale data, human-driven exploration*
  - Time series analysis – *repeatable scientific experiments to build confidence in ML*
  - Resource management – *variable cost analyses and experts*
  - Adversarial resilience – *detecting attempts to manipulate or evade detection*

Our Dataset

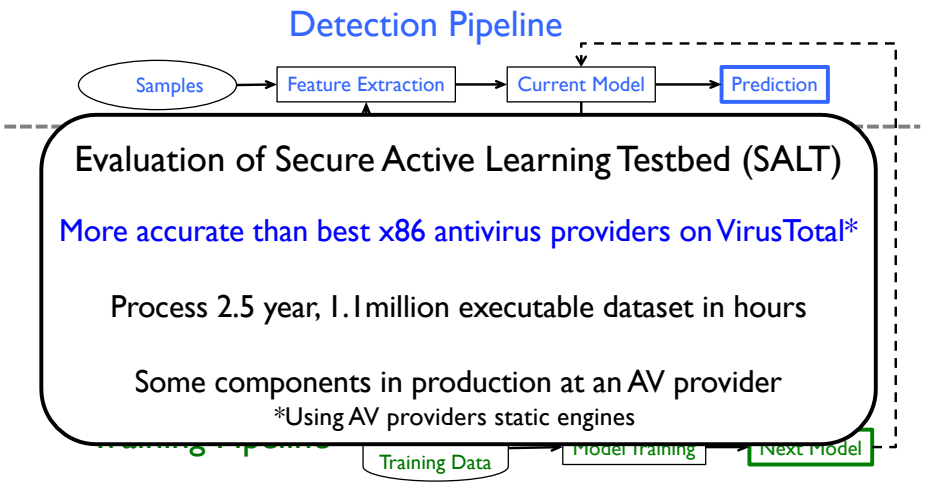
- Over 1 million unique executables from VirusTotal
  - Seen over 5 million times
  - Spanning January 2012 to June 2014
  - ~1% of VirusTotal's executables for the period
- Labeled by 32 AV providers' static engines
  - Threshold of four detections to label as malicious
- 85% malicious (3,000 to 406,000 families)
- *The largest academic malware detection dataset*

Secure Active Learning Testbed (SALT)



Built on Apache Spark for scalability

Secure Active Learning Testbed (SALT)



Evaluation of Secure Active Learning Testbed (SALT)

More accurate than best x86 antivirus providers on VirusTotal*

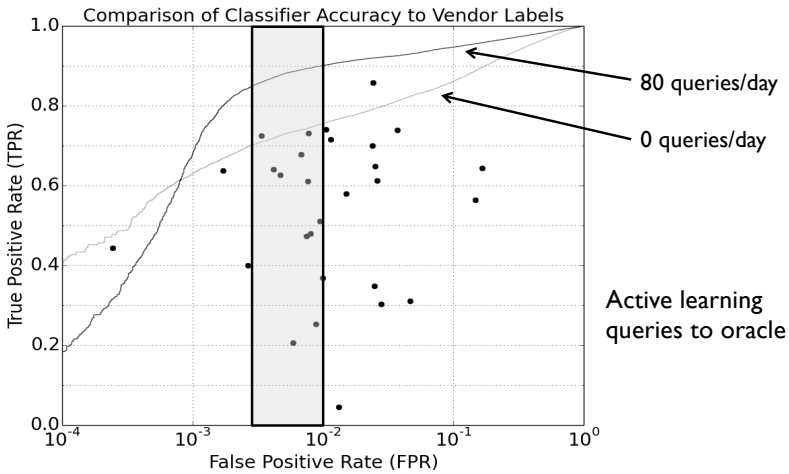
Process 2.5 year, 1.1million executable dataset in hours

Some components in production at an AV provider

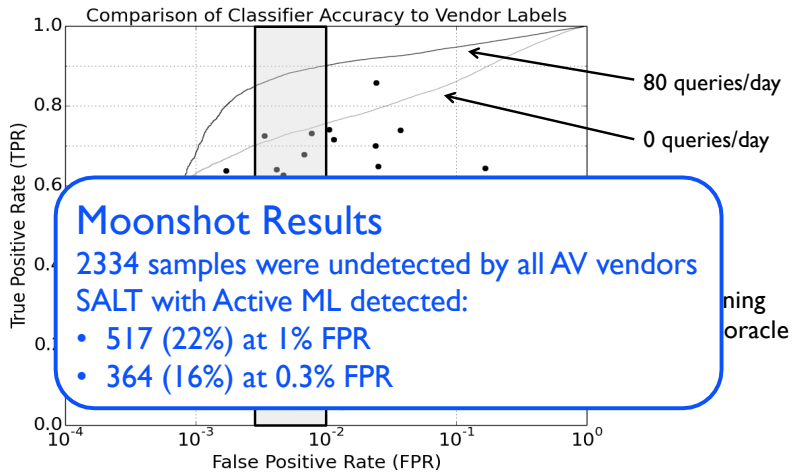
*Using AV providers static engines

Built on Apache Spark for scalability

Active Learning Results



Active Learning Results



Moonshot Results

2334 samples were undetected by all AV vendors

SALT with Active ML detected:

- 517 (22%) at 1% FPR
- 364 (16%) at 0.3% FPR

Current Research: Features

- Understanding feature importance
  - Most important features for detecting malware
  - Most important features for detecting benign
- Understanding important feature sensitivity
  - How brittle are important features?
  - Can features be manipulated to cause FN/FP errors?
- Some challenges:
  - How to change features without breaking malware?
  - How to make models robust?

Summary: Secure Machine Learning Goals

- Developing tools combining machine learning and analysis to automatically extract features and build models
- Improving users' experiences by translating the reasoning behind security decisions into human understandable concepts
- Designing robust algorithms for large-scale machine-learning in the presence of adversarial manipulation

SecML Acknowledgements

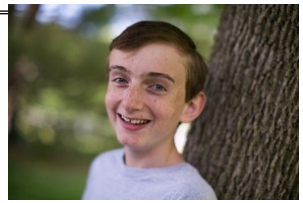
- **UC Berkeley:** David Fifield, Alex Kantchelian, Brad Miller, Qi Zhong, Vaishaal Shankar, Tony Wu, George Yu, Riyaz Faizullabhoy, Doug Tygar, Anthony D. Joseph
- **Drexel:** Eleanor Cawthon, Edwin Dauber, Rachel Greenstadt
- **ICSI:** Sadia Afroz, Michael Tshantz
- **Intel:** Ling Huang, Rekha Bachwani, Catherine Huang
- **Mitsubishi:** Takumi Yamamoto

Why Do We Need Big Genomic Data?

Precision Medicine!

- Understand genetic causes of diseases
- Choose more effective drugs and avoid adverse reactions

A Precision Medicine First



- Joshua Osborn (14 years old)
  - Visited ER 3 times in 4 months with encephalitis (brain swelling)
  - Hospitalized after 3rd visit and placed into a medically induced coma due to uncontrollable seizures
- Doctors tried traditional medicine approaches
  - Over 100 viral/fungal/bacterial pathogens cause encephalitis
  - Lots and lots of tests for different pathogens
  - 1 cm^3 of brain tissue biopsied
- Last resort – sent spinal fluid to UCSF for DNA sequencing
  - Needle in a haystack – throw out human reads (99.984%)
  - 475 reads out of 3M from rare *Leptospira santarosai* bacteria → given antibiotics, out of coma in week, discharged 4 wks

The NEW ENGLAND JOURNAL of MEDICINE

HOME | ARTICLES & MULTIMEDIA | ISSUES | SPECIALTIES & TOPICS | FOR AUTHORS | CME

ORIGINAL ARTICLE
BRIEF REPORT June 4, 2014

Actionable Diagnosis of Neuroleptospirosis by Next-Generation Sequencing

Michael R. Wilson, M.D., Samia N. Naccache, Ph.D., Erik Samayoa, B.S., C.L.S., Mark Biagtan, M.D., Hiba Bashir, M.D., Guixia Yu, B.S., Shahriar M. Salamat, M.D., Ph.D., Sneha Somasekar, B.S., Scot Federman, B.A., Steve Miller, M.D., Ph.D., Robert Sokolic, M.D., Elizabeth Garabedian, R.N., M.S.L.S., Fabio Candotti, M.D., Rebecca H. Buckley, M.D., Kurt D. Reed, M.D., Teresa L. Meyer, R.N., M.S., Christine M. Seroogy, M.D., Renee Galloway, M.P.H., Sheryl L. Henderson, M.D., Ph.D., James E. Gern, M.D., Joseph L. DeRisi, Ph.D., and Charles Y. Chiu, M.D., Ph.D.

June 4, 2014 | DOI: 10.

Abstract

More than half
laboratory test
establishing a
can be difficult
discover emer
generation se
largely unexpl
bacterial caus
contributed di
outcome.

CASE REPORT

A 14-year-old boy with severe combined immunodeficiency (SCID) caused by adenosine deaminase deficiency and partial immune reconstitution after he had undergone two haploidentical bone marrow transplantations initially presented to the emergency department in early April 2013 after having had headache and fevers, with temperatures up to 39.4°C, for 6 days (Figure 1A). He was admitted to the hospital and discharged 1 day later after resolution of his fever and headache.

The patient's outpatient medications included monthly infusions of intravenous immune globulin for hypogammaglobulinemia and trimethoprim–sulfamethoxazole or atovaquone for prophylaxis against *Pneumocystis jirovecii* pneumonia. He had no known sick contacts but did have three pet cats. He had gone on a missionary trip to Puerto Rico during the first 2 weeks

FIGURE 1

Clinical Course of the

The New York Times

In a First, Test of DNA Finds Root of Illness

By CARL ZIMMER JUNE 4, 2014

Joshua Osborn, 14, lay in a coma at American Family Children's Hospital in Madison, Wis. For weeks his brain had been swelling with fluid, and a battery of tests had failed to reveal the cause.

The doctors told his parents, Clark and Julie, that they wanted to run one more test with an experimental new technology. Scientists would search Joshua's cerebrospinal fluid for pieces of DNA. Some of them might belong to the pathogen causing his encephalitis.

The Osborns agreed, although they were skeptical that the test would succeed where so

- US: 15,000 cases a year of encephalitis with 2,000 deaths (>70% of deaths underdiagnosed)
- Worldwide: 70,000 diagnosed cases a year with 25,000 deaths

only might it provide speedy diagnoses to critically ill patients, they said, it could lead to more effective treatments for maladies that can be hard to identify, such as Lyme disease.

Diagnosis is a crucial step in medicine, but it can also be the most difficult. Doctors usually must guess the most likely causes of a medical problem and then order individual tests to see which is the right diagnosis.

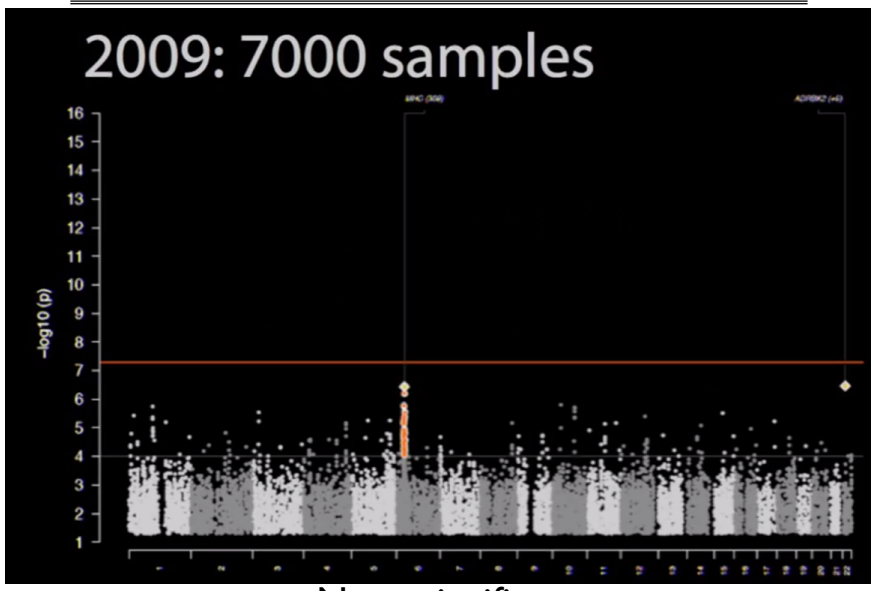
The guessing game can waste precious time. The causes of some conditions, like encephalitis, can be so hard to diagnose that doctors often end up with no answer at all.

"About 60 percent of the time, we never make a diagnosis" in encephalitis, said Dr. Michael R. Wilson, a neurologist at the University of California, San Francisco, and an author of the new paper. "It's frustrating whenever someone is doing poorly, but it's especially frustrating when we can't even tell the parents what the hell is going on."

y have been working on methods for 2003 Dr. Joseph DeRisi, a biochemist at

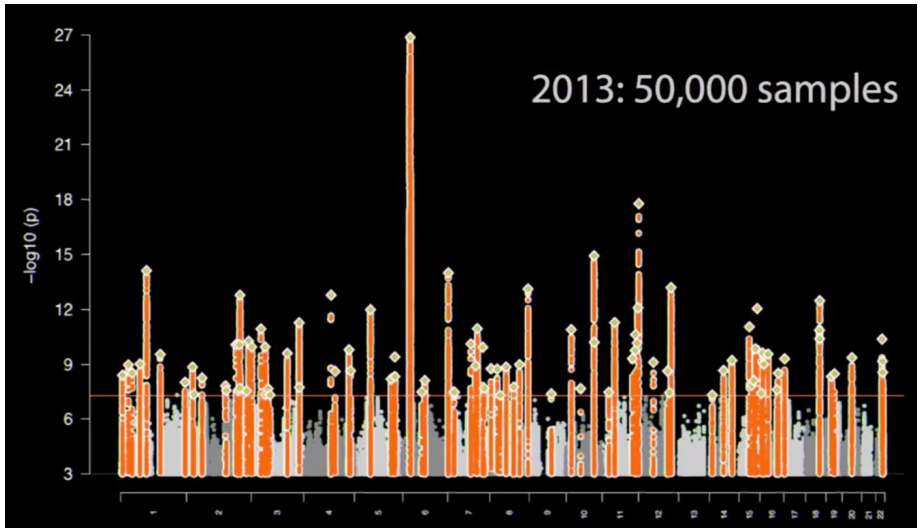
https://amplab.cs.berkeley.edu/snap-helps-save-a-life/

Looking for Genetic Causes of Schizophrenia



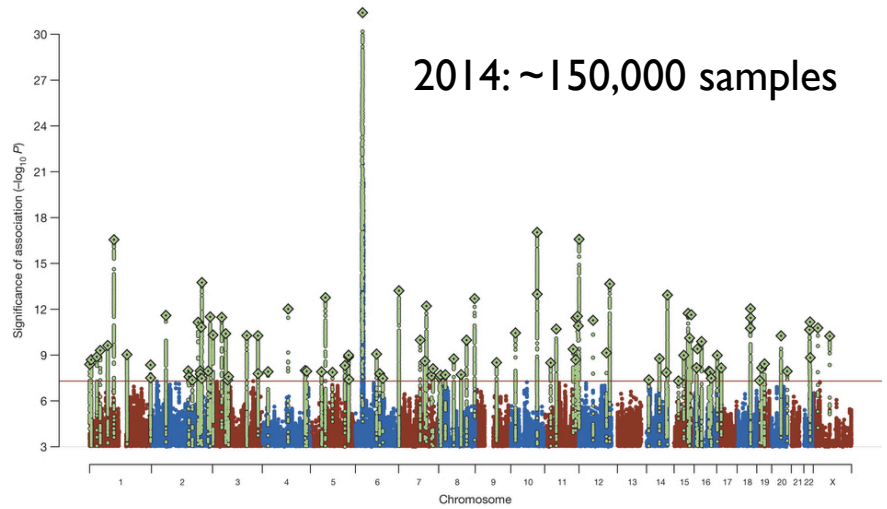
None significant

Looking for Genetic Causes of Schizophrenia



60 significant

Looking for Genetic Causes of Schizophrenia

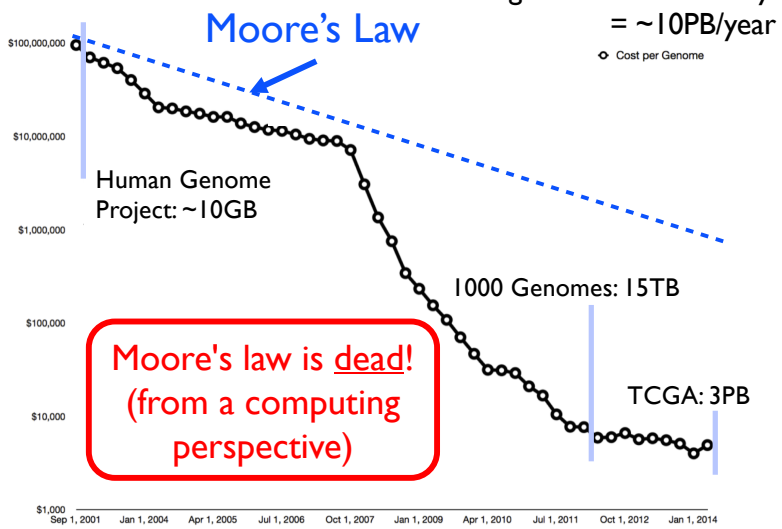


128 significant

Wet Lab Sequencing Cost Dropping Exponentially

Source: NIH National Genome Research Institute

2014: ~230,000 genomes sequenced
15-250GB/genome = ~30TB/day
= ~10PB/year



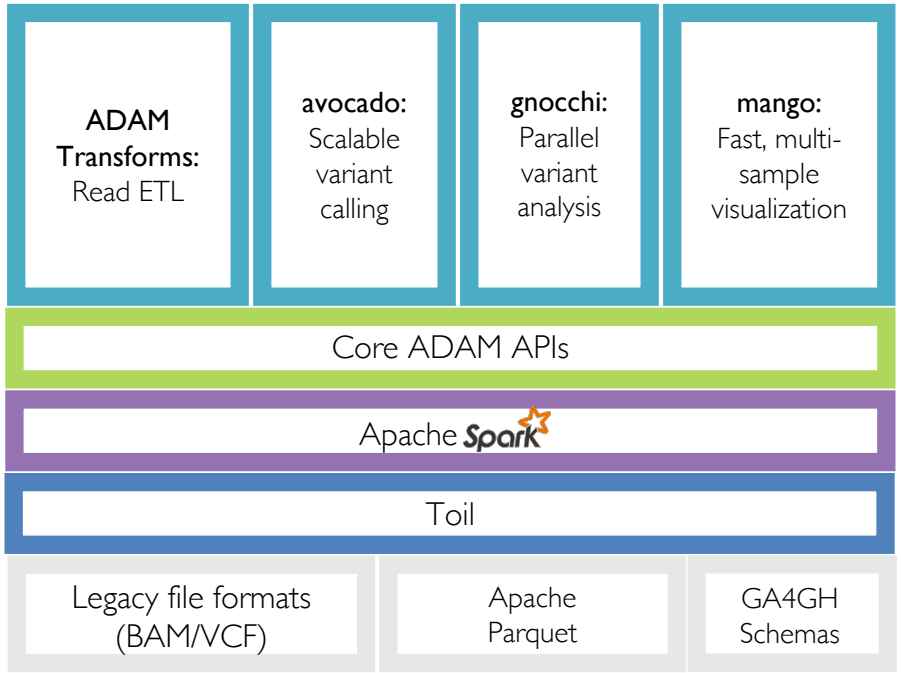
Moore's law is dead! (from a computing perspective)

Time to Break from Legacy Compute!

- Large cohorts generate massive amounts of data
  - 100+ GB of sequence data per sample
  - 1+ GB of variant data per sample
  - US Veterans Administration's Million Veterans Project (210 PB)
    - » Enrolled 500,000th veteran in August 2016
  - UK 100K Genome Project (21 PB)
    - » One goal is to analyze rare diseases (<1 in 2,000 people)
- Historically, we have processed genomic data on HPC
  - High Performance Computing relies on fast, expensive machines with centralized storage → expensive to scale data volume

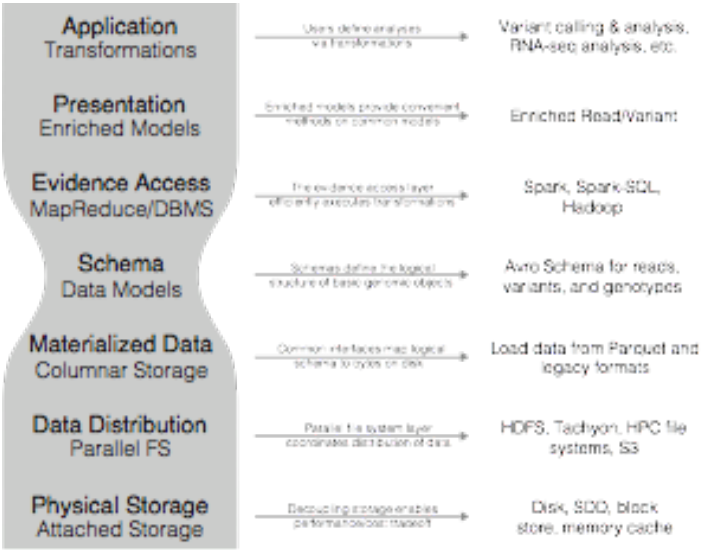
Our Approach

- Bring together cross cutting researchers
  - We have CS, Bio, HPC backgrounds
- Leverage fact that most genomic analysis steps are parallel
- Rethink design of genomics tools for cloud/commodity computing
  - Easy to write and use horizontally scalable genomics algorithms
- Apply Open Source Software ethos
  - We build tools that people freely use and build upon
- Bring emphasis on collaboration:
  - Work closely with UCSC CGL, MSR, OHSU, Stanford, Mt. Sinai



ADAM: A Stack-based Design

A Well Designed Stack Simplifies Application Design



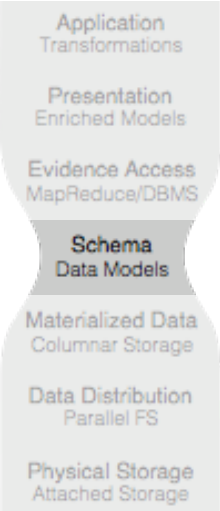
First, Define a Schema

```
record AlignmentRecord {
    union { null, Contig } contig = null;
    union { null, long } start = null;
    union { null, long } end = null;
    union { null, int } mapq = null;
    union { null, string } readName = null;
    union { null, string } sequence = null;
    union { null, string } mateReference = null;
    union { null, long } mateAlignmentStart = null;
    union { null, string } cigar = null;
    union { null, string } qual = null;
    union { null, string } recordGroupName = null;
    union { int, null } basesTrimmedFromStart = 0;
    union { int, null } basesTrimmedFromEnd = 0;
    union { boolean, null } readPaired = false;
    union { boolean, null } properPair = false;
    union { boolean, null } readMapped = false;
    union { boolean, null } mateMapped = false;
    union { boolean, null } firstOfPair = false;
    union { boolean, null } secondOfPair = false;
    union { boolean, null } failedVendorQualityChecks = false;
    union { boolean, null } duplicateRead = false;
    union { boolean, null } readNegativeStrand = false;
    union { boolean, null } mateNegativeStrand = false;
    union { boolean, null } primaryAlignment = false;
    union { boolean, null } secondaryAlignment = false;
    union { boolean, null } supplementaryAlignment = false;
    union { null, string } mismatchingPositions = null;
    union { null, string } origQual = null;
    union { null, string } attributes = null;
    union { null, string } recordGroupSequencingCenter = null;
    union { null, string } recordGroupDescription = null;
    union { null, long } recordGroupRunDateEpoch = null;
    union { null, string } recordGroupFlowOrder = null;
    union { null, string } recordGroupKeySequence = null;
    union { null, string } recordGroupLibrary = null;
    union { null, int } recordGroupPredictedMedianInsertSize = null;
    union { null, string } recordGroupPlatform = null;
    union { null, string } recordGroupPlatformUnit = null;
    union { null, string } recordGroupSample = null;
    union { null, Contig } mateContig = null;
}
```

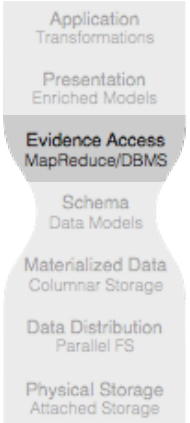
Schema Data Models

Query Genomic Data via Schema

```
record AlignmentRecord {
    union { null, Contig } contig = null;
    union { null, long } start = null;
    union { null, long } end = null;
    union { null, int } mapq = null;
    union { null, string } readName = null;
    union { null, string } sequence = null;
    union { null, string } mateReference = null;
    union { null, long } mateAlignmentStart = null;
    union { null, string } cigar = null;
    union { null, string } qual = null;
    union { null, string } recordGroupName = null;
    union { int, null } basesTrimmedFromStart = 0;
    union { int, null } basesTrimmedFromEnd = 0;
    union { boolean, null } readPaired = false;
    union { boolean, null } properPair = false;
    union { boolean, null } readMapped = false;
    union { boolean, null } mateMapped = false;
    union { boolean, null } firstOfPair = false;
    union { boolean, null } secondOfPair = false;
    union { boolean, null } failedVendorQualityChecks = false;
    union { boolean, null } duplicateRead = false;
    union { boolean, null } readNegativeStrand = false;
    union { boolean, null } mateNegativeStrand = false;
    union { boolean, null } primaryAlignment = false;
    union { boolean, null } secondaryAlignment = false;
    union { boolean, null } supplementaryAlignment = false;
    union { null, string } mismatchingPositions = null;
    union { null, string } origQual = null;
    union { null, string } attributes = null;
    union { null, string } recordGroupSequencingCenter = null;
    union { null, string } recordGroupDescription = null;
    union { null, long } recordGroupRunDateEpoch = null;
    union { null, string } recordGroupFlowOrder = null;
    union { null, string } recordGroupKeySequence = null;
    union { null, string } recordGroupLibrary = null;
    union { null, int } recordGroupPredictedMedianInsertSize = null;
    union { null, string } recordGroupPlatform = null;
    union { null, string } recordGroupPlatformUnit = null;
    union { null, string } recordGroupSample = null;
    union { null, Contig } mateContig = null;
}
```

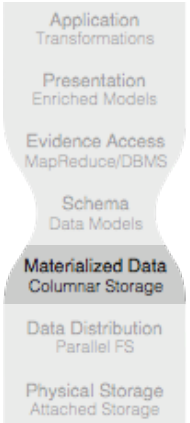


Parallelize Common Access Patterns



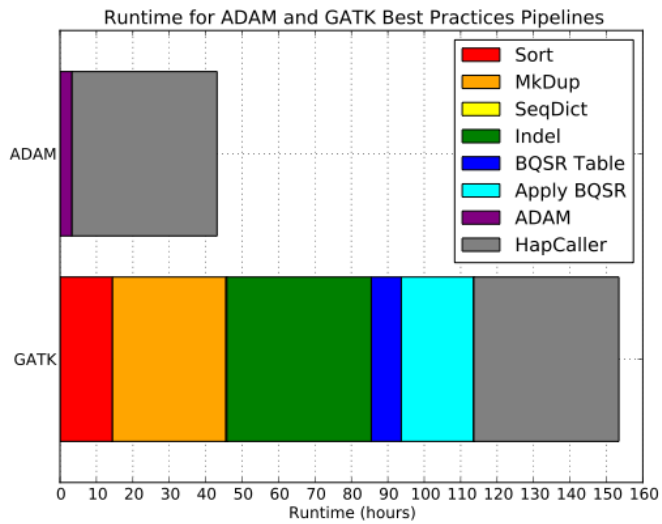
- We provide parallel methods for processing genomic data with Spark
- Makes use of genomics specific reference genome structure to accelerate queries
- E.g., region join for high performance overlap queries

Pick Appropriate Storage



- When accessing scientific datasets, we frequently slice and dice the dataset:
  - Algorithms may touch subsets of columns
  - We don't always touch the whole dataset
- This is a good match for columnar storage like Apache Parquet
- *Also, can support legacy formats*

Big Performance Gains!



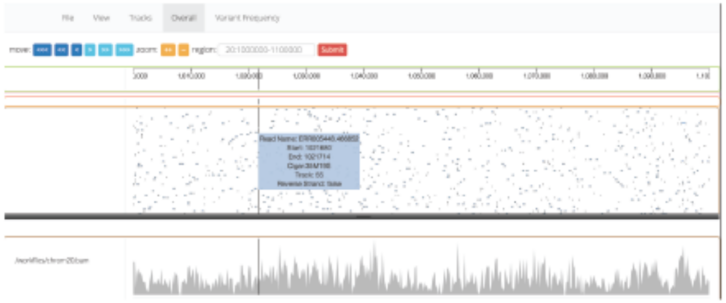
ADAM-based pipeline is statistically identical to the (non-OSS) GATK, but 3.5x faster, 3x cheaper

Cloud-scale Batch Variant Analysis

- *Resequencing*: sequence a sample, and compute `diff` from "average" genome → variants
- Projects: Toil, ADAM, avocado
- Algorithmic challenges:
  - Empirical evaluation of preprocessing methods
  - Use of alt haplotypes/graph genomes in variant calling
  - Linear time genome reassembly algorithms
- Infrastructure challenges:
  - How do we run geo-distributed workflows efficiently?
  - Can bioinformatics be made reproducible?

Exploratory Data Analysis for Genomics

- Projects: mango
- Currently, exploring genomic data means:
  - `cat my.vcf | grep "chr1" | awk '…' | sed …`
- Viz tools can only explore short ranges
- Let us break down this wall! Long range viz with ad hoc queries



Scalable ML for Genomics

- Projects: gnocchi, fig, …
- Genomic data is often statistically interesting → need to train some higher level classifier
  - Genome-Wide Association Studies (GWAS)
  - Phenome-Wide Association Studies (PheWAS)
- Spark is designed for ML at scale; how can we harness this for genomics?
  - GWAS/PheWas/population genomics → Gnocchi
  - Non-coding variant annotation → fig

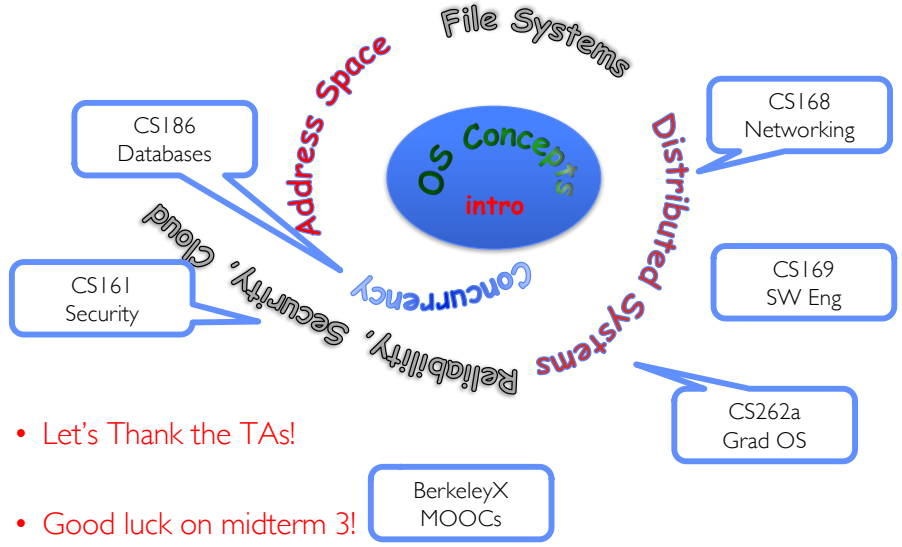
Want to Learn More?

- Check out the code!
  - ADAM: https://github.com/bigdatagenomics/adam
- Check out a demo!
  - https://databricks.com/blog/2016/05/24/genome-sequencing-in-a-nutshell.html
- Run ADAM in Databricks Community Edition!
  - http://goo.gl/xK8x7s

Acknowledgements

- **UC Berkeley:** Matt Massie, Timothy Danford, André Schumacher, Jey Kottalam, Karen Feng, Eric Tu, Niranjan Kumar, Ananth Pallaseni, Michael Heuer, Alyssa Morrow, Frank A. Nothaft, Justin Paschall, Anthony D. Joseph, Dave Patterson
- **Mt. Sinai:** Arun Ahuja, Neal Sidhwaney, Ryan Williams, Michael Linderman, Jeff Hammerbacher
- **GenomeBridge:** Carl Yeksigian
- **Cloudera:** Uri Laserson, Tom White
- **Microsoft Research:** Ravi Pandya, Bill Bolosky
- **UC Santa Cruz:** Benedict Paten, David Haussler, Hannes Schmidt, Beau Norgeot, Audrey Musselman-Brown, John Vivian, Jacob Pfeil
- And many other open source contributors, especially Neil Ferguson, Andy Petrella, Xavier Tordior
- Total of >50 contributors to ADAM/BDG from >12 institutions

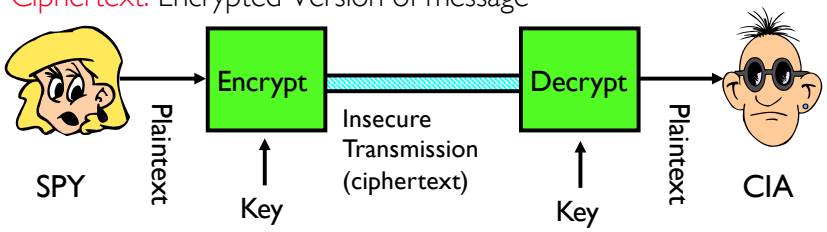
Thank you!



- Let's Thank the TAs!
- Good luck on midterm 3!

Recall: Private Key Cryptography

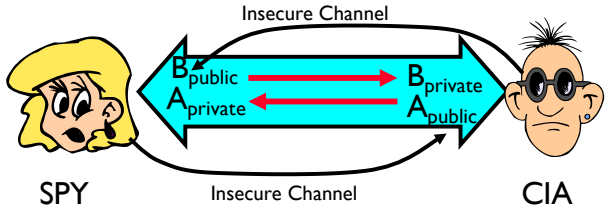
- Private Key (Symmetric) Encryption:
  - Single key used for both encryption and decryption
- Plaintext: Unencrypted Version of message
- Ciphertext: Encrypted Version of message



- Important properties
  - Can't derive plain text from ciphertext (decode) without access to key
  - Can't derive key from plain text and ciphertext
  - As long as password stays secret, get both secrecy and authentication
- Symmetric Key Algorithms: DES, Triple-DES, AES

Recall: Public Key Encryption Details

- Idea: K_{public} can be made public, keep $K_{private}$ private



- Gives message privacy (restricted receiver):
  - Public keys (secure destination points) can be acquired by anyone/used by anyone
  - Only person with private key can decrypt message
- What about authentication?
  - Use combination of private and public key
  - Alice→Bob: $[(\text{I'm Alice})^{Aprivate}\ \text{Rest of message}]^{Bpublic}$
  - Provides restricted sender and receiver
- But: how does Alice know that it was Bob who sent her B_{public}? And vice versa…
  - Need a certificate authority to sign keys!