ECEN 5053-002

Developing the Industrial Internet of Things

Week 1

Expectations and Course Overview

Dave Sluiter - Spring 2018

Be Boulder.

University of Colorado **Boulder**

Expectations and Course Overview

Who Am I?

- 30+ year chip design engineer
- Unisys (Sperry Univac) - Mainframe floating point unit
- Artist Graphics - 3D graphics chips
- LSI Logic -
  - North American Field Coreware Manager - MIPS CPUs, CoreWare Methodology
  - Lead hardware architect for satellite set-top box decoder chips
- Seagate/Micron - Hardware architect for enterprise solid state drives

Logistics

- Communication:
  - D2L
  - Email

Expectations

- You are Graduate students!
- Attendance/Participation is important
- Honor Code Violations:
  - **Do not** represent someone else's work as your own
  - I **do want** collaboration and exchange of ideas
  - I **do not want** copying - Do your own work!
  - As graduate students, you will not be given a 2nd chance

Where does this course fit?

ECEN 5053-002
Developing the Industrial Internet of Things

ESE Core and Elective Courses

Many new topic areas

Course Overview

- Format
  - Traditional lecture
- Survey Course
  - Intention is to introduce you to a wide set of subject matter
  - Broaden your scope/viewpoint/vocabulary
- Assessments
  - Attendance/Participation
  - Small number of projects - easy, provides hands-on experience
  - Midterm exam
  - Final exam
- No homework 

Course Overview (con't)

- Primary focus
  - Emerging trends: Market and Technical
  - Key business concepts for engineers
  - Key skills to develop
  - Understanding the "big picture", how these systems are built and the value propositions they offer
  - Your role in DIIoT - Show you how your ESE core and elective courses intersect with IIoT

Course Overview (con't)

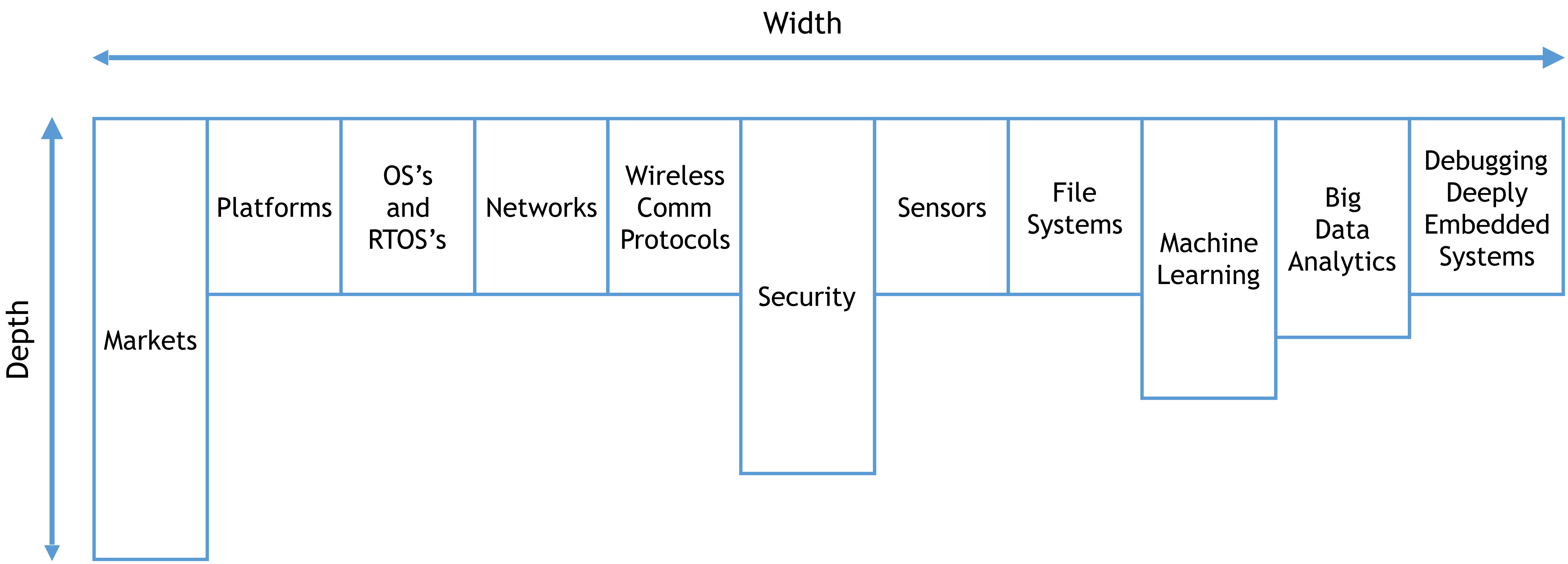
- What is Industrial IoT?

IoT Market Spectrum

Not Industrial	**Partial Industrial**	**Industrial**			
Consumer Electronics	Retail	Building Automation	Agriculture	Automotive & Transportation	Manufacturing
	Healthcare	Oil & Gas		Utilities	

Course

Topic Areas



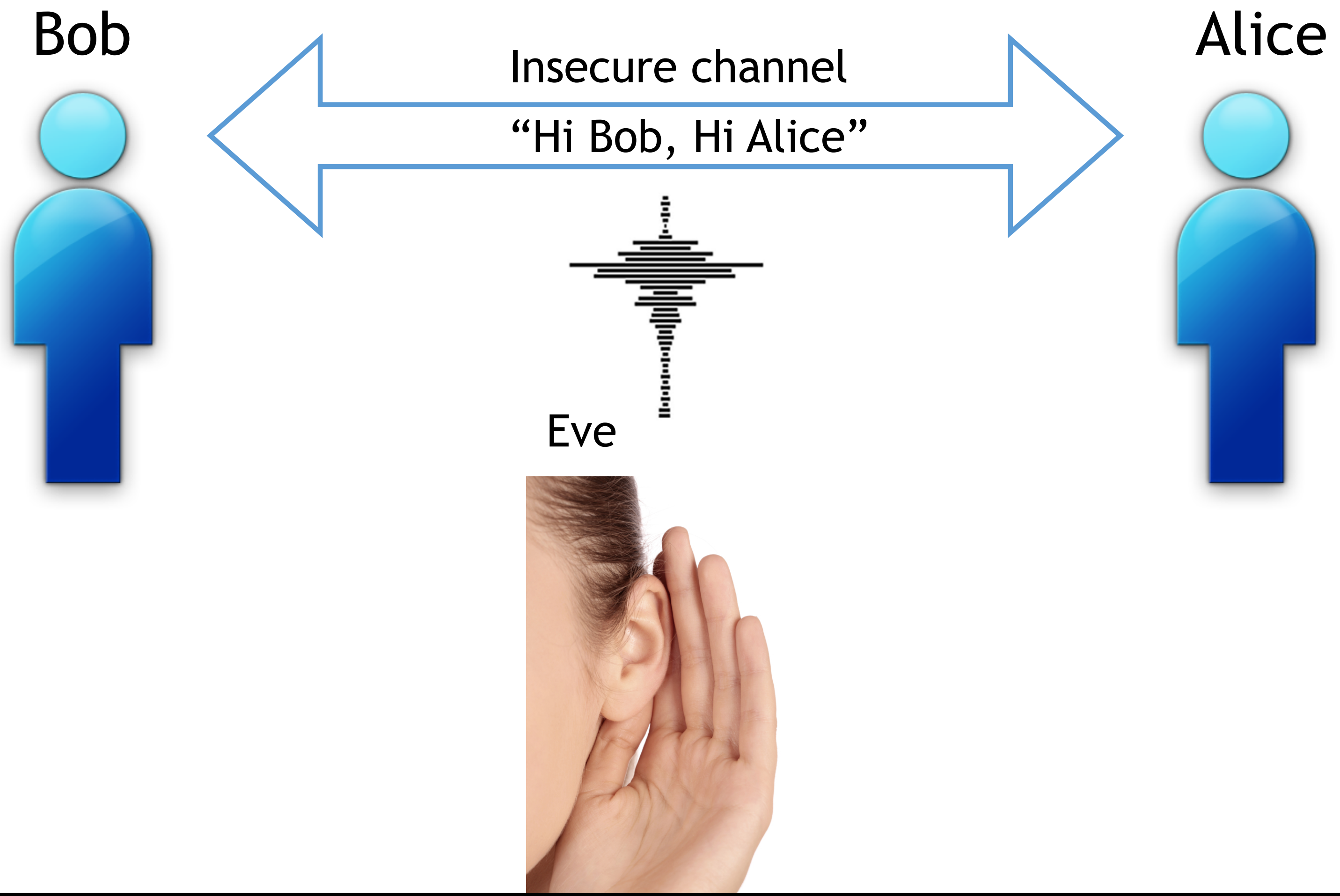
Width
Depth
Markets
Platforms
OS's and RTOS's
Networks
Wireless Comm Protocols
Security
Sensors
File Systems
Machine Learning
Big Data Analytics
Debugging Deeply Embedded Systems

What will be covered

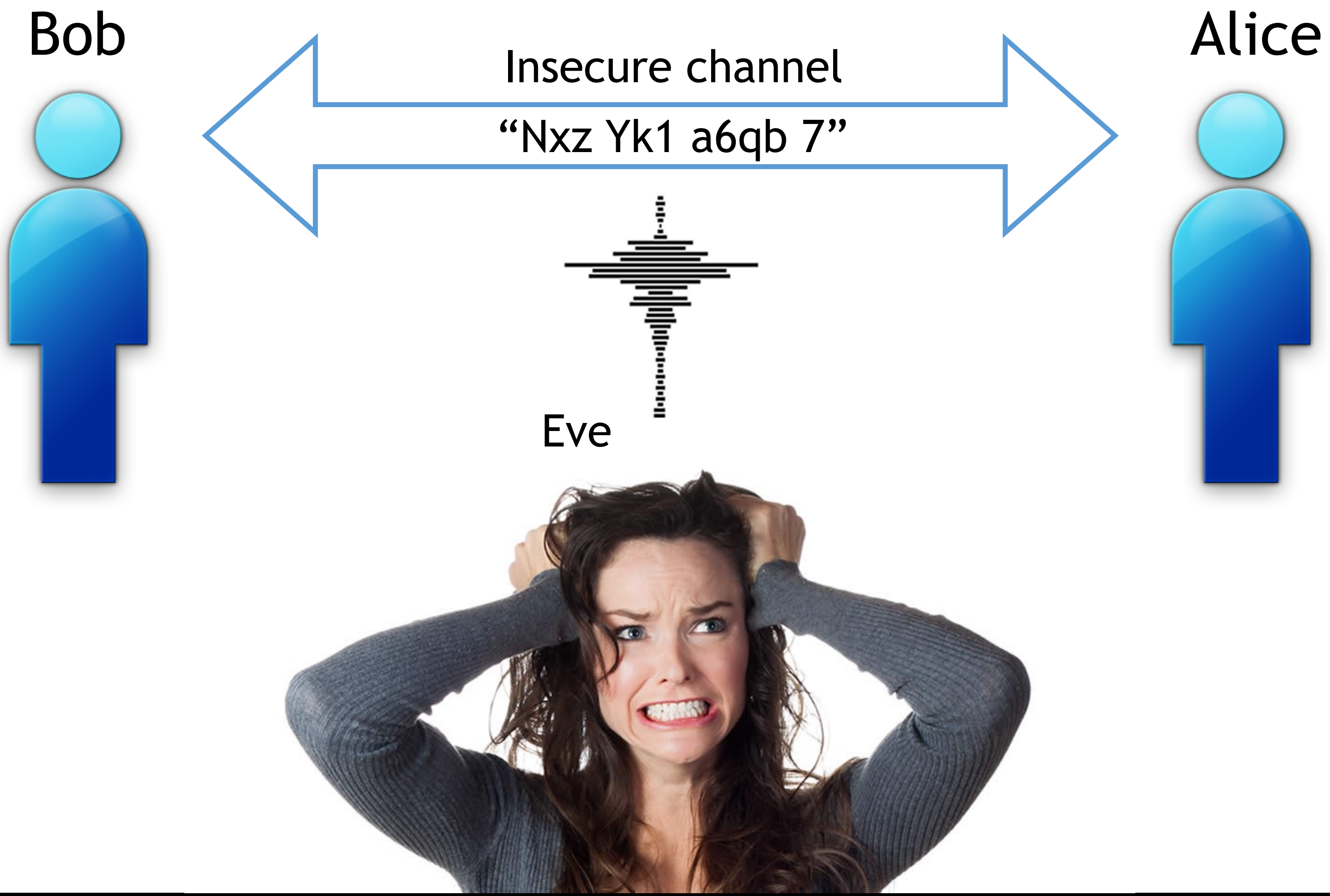
- An inside look at markets; size and $ estimates
- Platforms, IBM Bluemix + Watson
- Networks, NFV and SDN
- Security
- Project planning, staffing and execution
- Sensors, file systems
- Machine learning
- Big data analytics
- SystemC
- Debugging deeply embedded systems
- Guest speakers

Security

What does it mean to be secure?

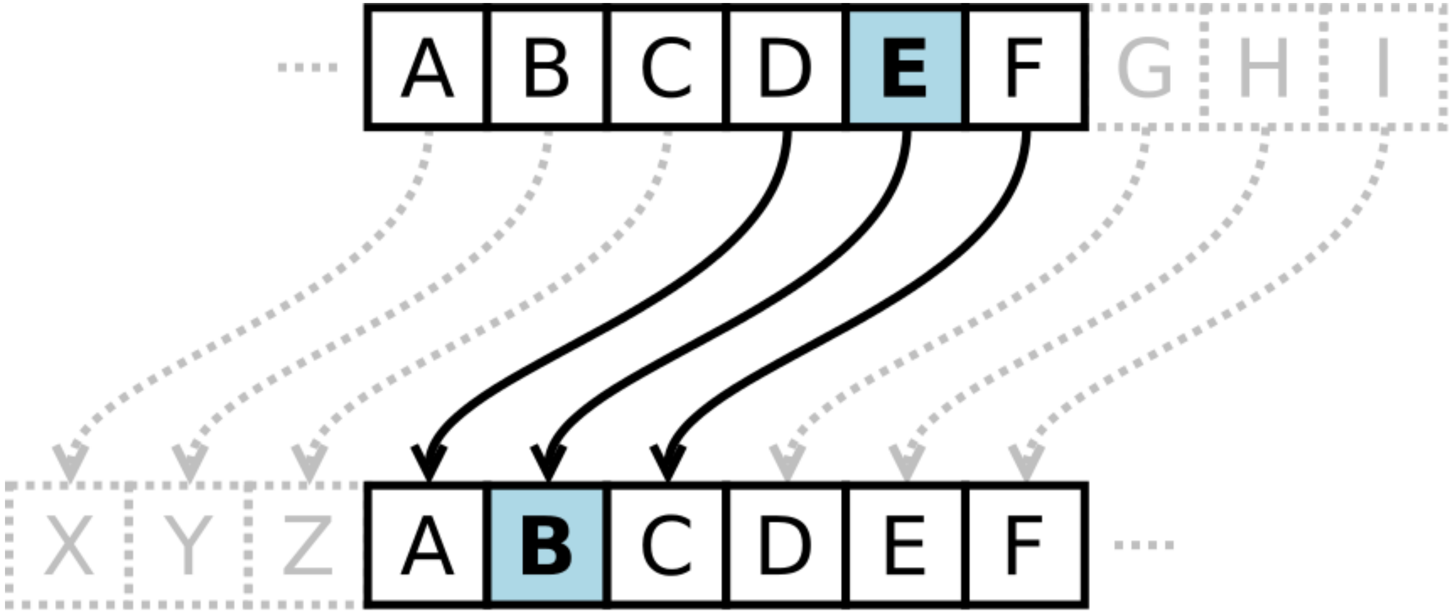


What does it mean to be secure?



Encryption Techniques

- Caesar Cipher, named after Julius Caesar
- Substitution cipher, based on a shift



Encryption Techniques

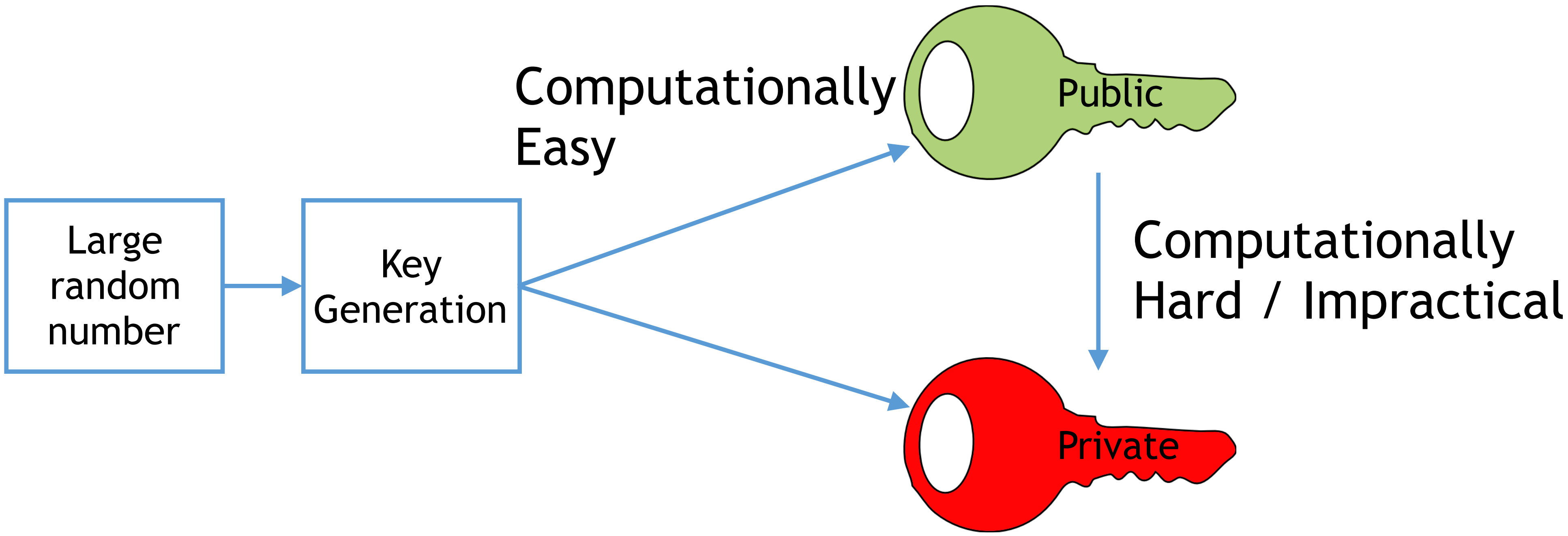
- One time pad (OTP)
- So-called “perfect” encryption
- Impractical for real-world

A=0, B=1, C=2, … Z=25											
Plain text	M	E	E	T	T	O	N	I	G	H	T
	12	4	4	19	19	14	13	8	6	7	19
Key	D	Z	H	S	U	I	M	W	E	K	C
	3	25	7	18	20	8	12	22	4	10	2
Sum	15	29	11	37	39	22	25	30	10	17	21
Sum mod 26	15	3	11	11	13	22	25	4	10	17	21
Cipher Text	P	D	L	L	N	W	Z	E	K	R	V
Cipher text	P	D	L	L	N	W	Z	C	K	R	V
	15	3	11	11	13	22	25	4	10	17	21
Key	D	Z	H	S	U	I	M	W	E	K	C
	3	25	7	18	20	8	12	22	4	10	2
Diff	12	-22	4	-7	-7	14	13	-18	6	7	19
Sum mod 26	12	4	4	19	19	14	13	8	6	7	19
Plain text	M	E	E	T	T	O	N	I	G	H	T

AES (Advanced Encryption Standard)

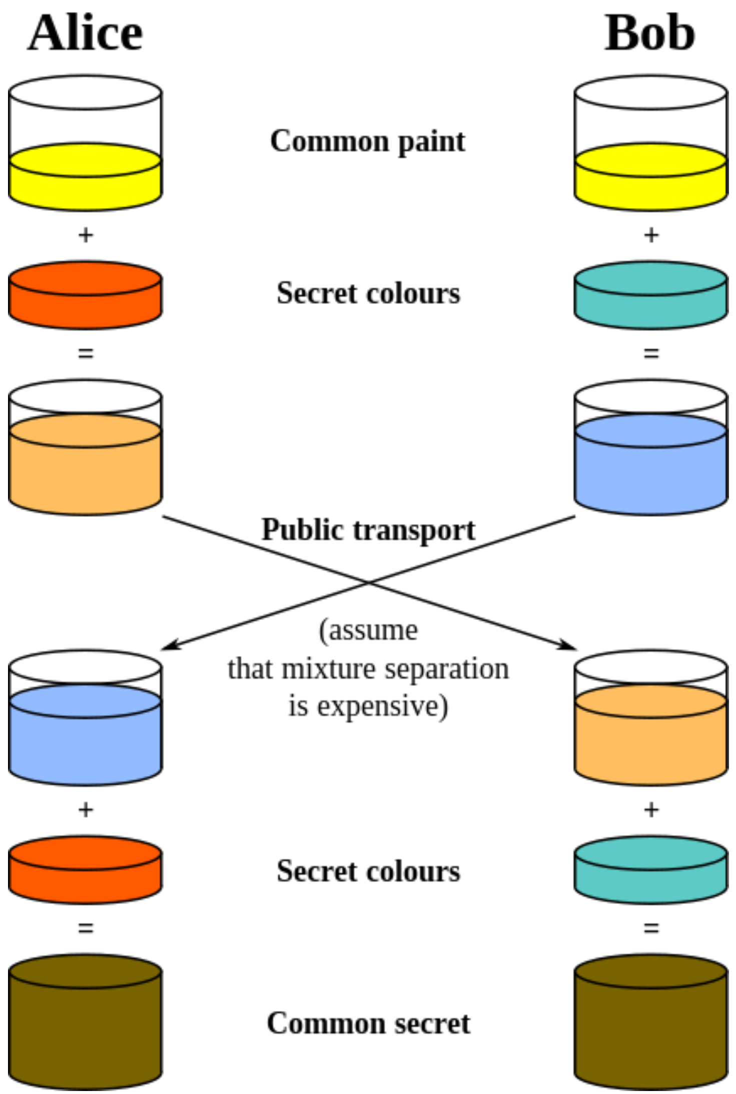
- Established by US NIST (National Institute of Standards)
- Block cipher: 16-bytes in, 16-byte out
- 3 key lengths: 128-, 192-, 256-bits
- High-level description
  - 1) With N=number of rounds, round keys are extracted from the cipher key (where N = 10, 12, 14, for 128-, 192- or 256-bits)
  - 2) Round 0
  - 3) Rounds 1 to N-2
  - 4) Final round N-1
- Believed to be secure, but we don’t how to prove it.

Asymmetric Encryption



Diffie-Hellman

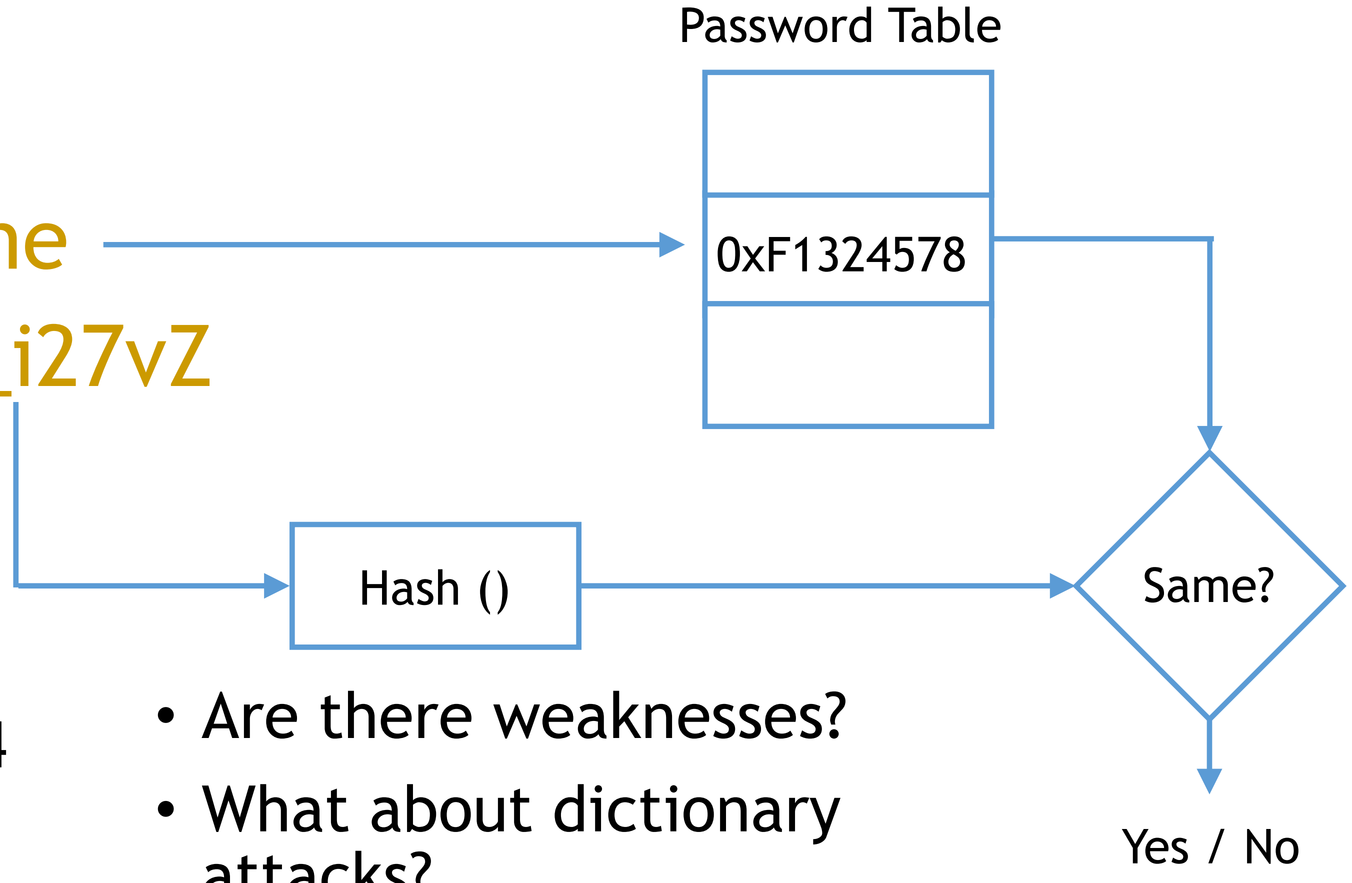
A method to securely establish a known secret (a "key") between 2 parties over an insecure channel.



Source: https://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange

Uses for Hash Functions

- Saving passwords
  - Login: UserName
  - Password: Xh8_i27vZ



- See also the SHA-2 family of hash functions as per FIPS 180-4
- Well studied, haven’t spotted a problem yet

- Are there weaknesses?
- What about dictionary attacks?

Take Aways

- Security Mind-set
- Data Integrity
- Authentication
- Encryption
- Will look at few “security blunders”

Machine Learning

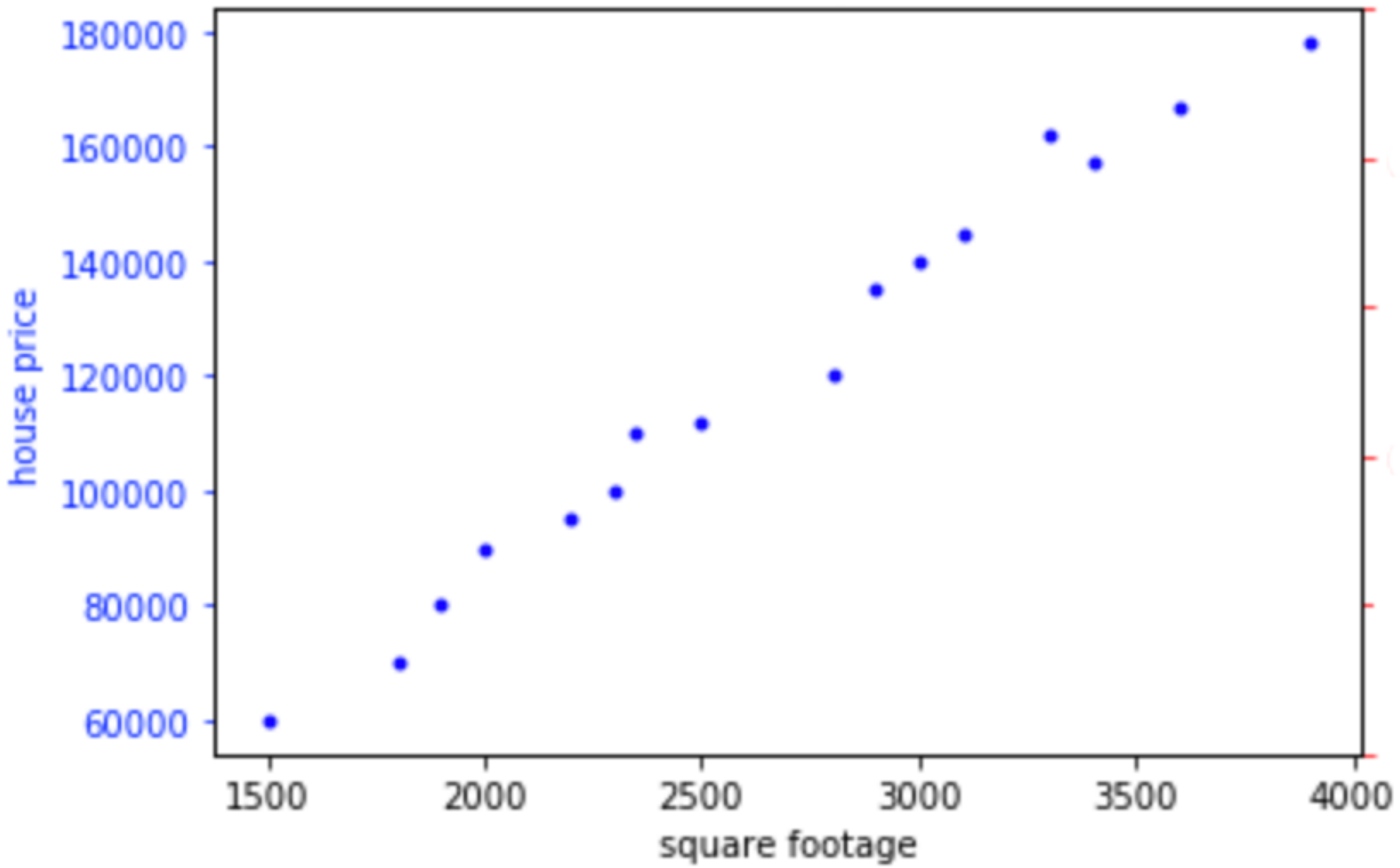
Linear Regression

- We have some example data
    - Training data
- Each row is an example from real sales data
- Contains a number of features, x_n
    - x_1 is square footage
    - x_2 is number of bedrooms
- And output y, the price
    - Is what we are trying to predict, also known as the **target value**

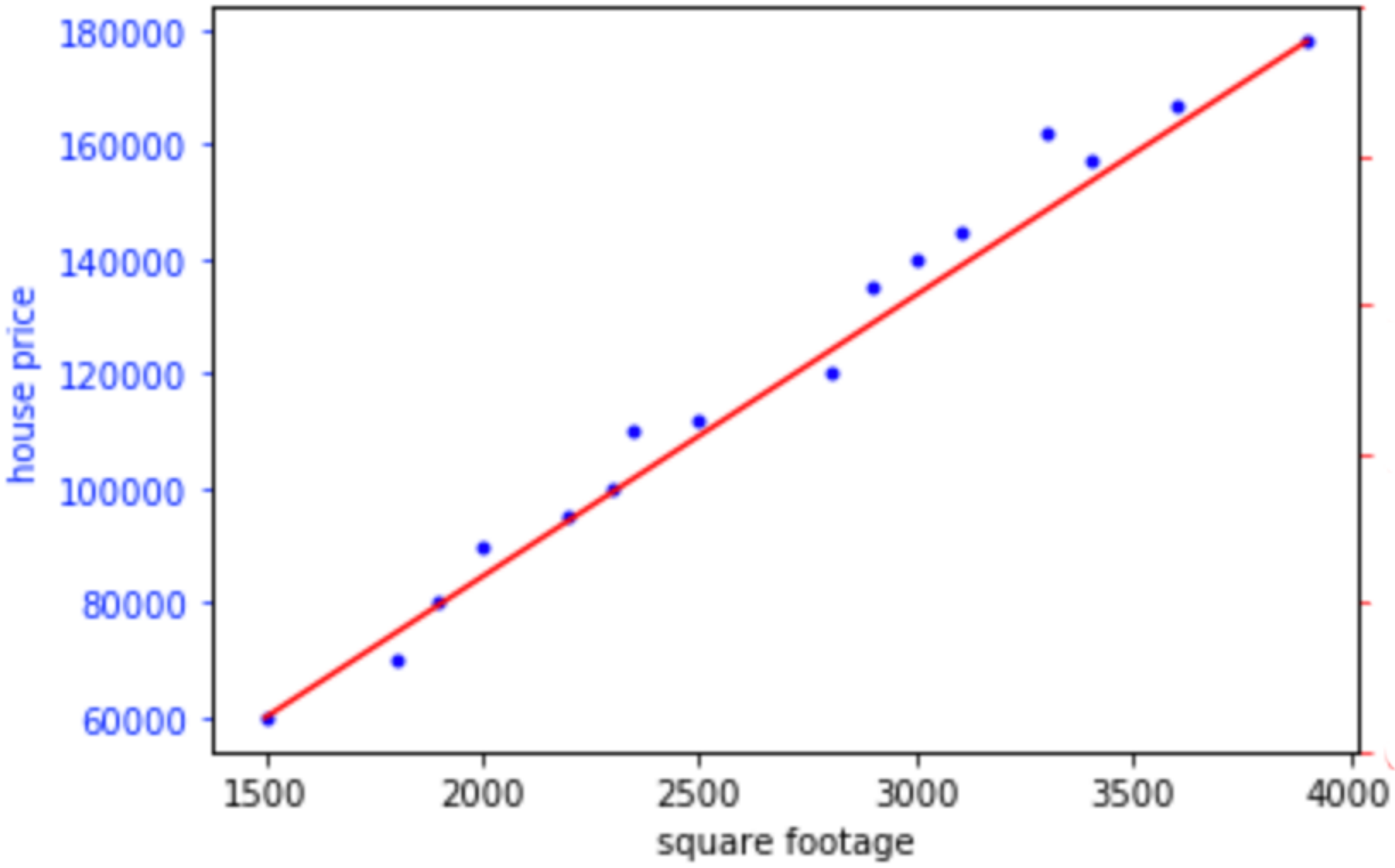
```
train_data = np.array(
    [
    # sqft, #bedrooms, price
    [1500, 2,  60000],
    [1800, 2,  70000],
    [1900, 2,  80000],
    [2000, 3,  90000],
    [2200, 3,  95000],
    [2300, 2, 100000],
    [2350, 3, 110000],
    [2500, 3, 112000],
    [2800, 4, 120000],
    [2900, 3, 135000],
    [3000, 4, 140000],
    [3100, 4, 145000],
    [3300, 5, 162000],
    [3400, 4, 157000],
    [3600, 5, 167000],
    [3900, 5, 178000]
    ]
) # end training data
```

Training Data

of bedrooms is not plotted



We can manually draw a best-fit curve



Linear Regression

- We want to create a hypothesis function *h()* that will make predictions
- To perform supervised learning, we have to decide how we will represent the hypothesis function *h()*
- As an initial choice, let's say we decide to approximate *h()* as a linear combination of features x and weights θ, so we have:
  - $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2$
  - where we set $x_0 = 1$, and θ_0 becomes the intercept term
    - like in the equation for a line $y = ax + b$, b is the intercept term

Linear Regression

- m = number of training examples, 16
- n = number of features, 2
- We can rewrite $h_\theta(x)$ as :
  - $h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$

$$\boldsymbol{\theta} = [\theta_0 \quad \theta_1 \quad \theta_2 \quad \dots \quad \theta n] \qquad \boldsymbol{x} = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_n]$$

$$\theta^T x = [\theta_0, \theta_1, \theta_2 .. \theta_n][x_0 \quad x_1 \quad x_2 \quad \dots \quad x_n] = \mathbb{R} = h_\theta(x)$$

a real number

Linear Regression

- How do we pick/calculate/learn the values for the θ_i*'s*?
  - As a starting point, we can make $h()$ ~= y
- We can define a **cost function**:
  - $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$
  - The superscript *i's* refer to training examples, not raise to a power!
- We want to choose θ to minimize $J(\theta)$
- To do so, let's use a search algorithm that starts with some initial guess for θ, and repeatedly changes θ to make $J(\theta)$ smaller, until we converge at a minimum $J(\theta)$ value.

Linear Regression

- **Search algorithm = Gradient Descent**
  - Start with some initial guess at θ
  - and then repeatedly perform the update:
  - $\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta), j = 0 \ldots n$
    - this rule is called the LMS update rule (least mean squares rule)
    - α (alpha) is the learning rate
    - Great care needs to be taken choosing alpha

Linear Regression

- **Gradient Descent**
  - Calculating the derivative, $\frac{\partial}{\partial \theta_j} J(\theta)$ and for all n, we get:
  - *repeat until convergence* $\{$

    $$\theta_j = \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^i, j = 0 \ldots n$$
  - $\}$
    - for each θ_j cycle through all m training examples
  - This method is called **batch gradient descent**
  - This $J(\theta)$ equation is a convex quadratic function, and has only a global minimum. Beware of functions that may have local minima, because gradient descent could get "stuck" in a local minima.

Linear Regression

- **Gradient Descent**
  - Test for convergence is a **user defined function**
  - Downside to **batch gradient descent** is that every θ update needs to run through all m training samples
    - Computationally expensive for large training sets

IIoT Intro

Source: World Economic Forum: https://www.youtube.com/watch?v=8NGzrtK7eV0

End