CSE 535 - Mobile Computing Notes

Module 1: Mobile Computing System Models

System models for MC

Smartphones:

- A standalone system - much like desktop - has CPU, memory, Radio, GPU, battery
- Also intended to be an edge device
- Sensors help in context-aware computing
- IMU sensors - Accelerometer, Gyroscope, and Orientation sensors
- Physiological sensors - can measure health params eg heart rate

Smartphone + cloud

- Generally follows a client-server architecture
- Concerns: Security, availability, and reliability (wireless communication is unpredictable, server load impacts response time)
- TCP/IP is the best effort - prone to failures
- Bandwidth is limited - impacts communication time (computation time vs communication time tradeoff is always there)
- Offloading - do you want to do computing in the phone or server?

Fog servers:

- Cloud servers are far away. Fog servers are nearer. Eg: laptop on same wifi network.
- Advantages:
    - Lesser communication time
    - Supports multiple communication protocols (cloud only supports TCP/IP)
- Disadvantages:
    - Lesser resources
- Issues:
    - Not as fast as cloud. So application needs to be reoptimized
    - Type of apps supported. Eg: difficult to train ML models. Fog can’t access other users’ data. It's proprietary.

External Sensors:

- Adding external sensors to smartphone, cloud, Fog systems
- ECG, Brain sensors, FMRI, CGM sensors can be connected to our devices via various methods like Bluetooth, NFC etc

- Sensors are resource-constrained. Security is a major concern. They are resource-constrained so they don't use much security while sending raw data.
- Advantage:
    - They give context-rich data (Context is the status of the environment) which is helpful for knowledge discovery required for smart applications.

Cloudlets:

- Cloudlets are much smaller versions of cloud or fog. They may not be close necessarily. Eg: Other people's smartphones or any such devices.
- Adhoc cloud network - people voluntarily giving their partial computation. Different threads are executed on different phones. This is called volunteer computing.
- Peer to peer computation advantage - you help me, I help you.
- Eg application: Smart investigator: searching for missing person. Match the missing person's image to all images in people's phones.

Adaptation and Smartness in Mobile Computing

Adaptation

- Mobile applications adapt to gain a better user experience. Example: Phone should adapt to youtube streaming based on network. We can compromise video or audio quality. What to compromise depends on the type of content Eg: Jazz concert vs football match. So requirements are dynamic in such mobile app development.

What can we adapt:

- Data adaptation: Data fidelity and agility
- Adapting data means applying certain transformations based on data. Fidelity means is this data accurate and timely. Agility means can some also use this data and reach conclusions faster. Example: Location tracking. Three courses of location - GPS, Cellular, and wifi.
- GPS might not be always available. Clouds may prevent GPS. There is a tradeoff between timeliness and accuracy. This is where agility comes in. Cellular data may not be agile in heavy traffic scenarios but in light traffic, cellular is more agile. So agility helps break the dilemma of timeliness and accuracy.

Adaptation types

- System-level and user-level adaptation
- System-level adaption - doesn't query user. Eg: the auto-brightness sensor

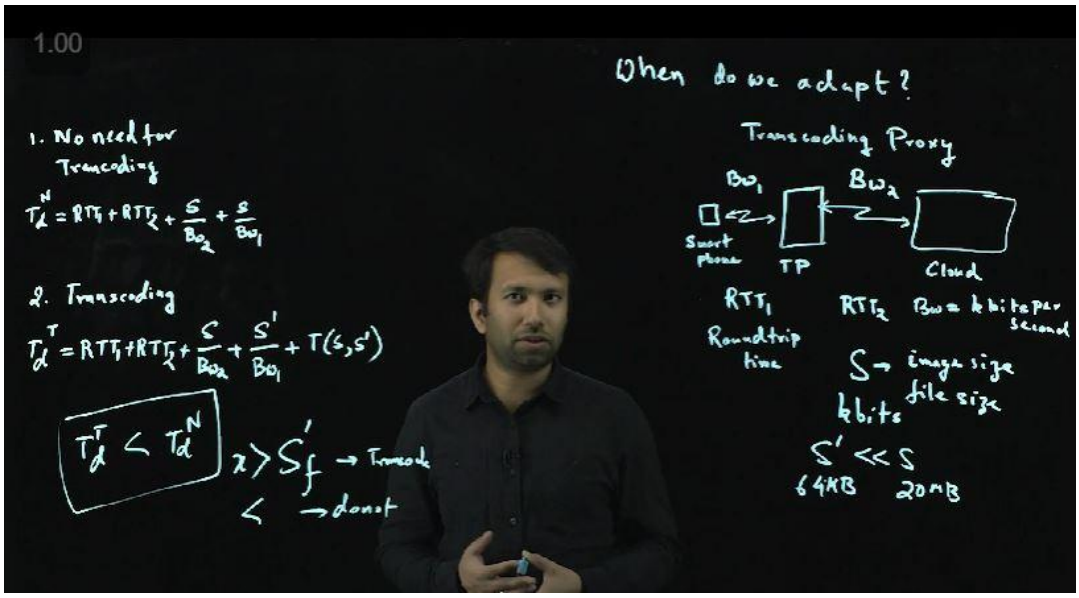
- User-level adaptation: The user is always asked. Eg youtube setting asking user to manually change video preference.
- Neither one of them alone suffice for a good user experience. System-level alone may be good enough in user-level adaptation may give cognitive overload. They are two ends of the spectrum.
- The balancing point between them is application-aware adaptation. That is what we will focus on. Eg: Consider content type when streaming youtube. Same Jazz vs Football match example above. IN this course we will try to achieve application-level adaptation and discuss its challenges.

Application-Aware adaptation types

- Reactive application-aware adaptation: Mobile app reacts to change in environment. First, there is a **change** then it is **sensed** and then reacts with the adaptation **decision**. Eg: Location-based restaurant search app - recommendation changes after location change. It is application-aware because it considers user preference also - you can tell what cuisine may be.
- Proactive application-aware adaptation: take adaptation decisions ahead of time
- We need predictive models. Eg: Gas buddy application. It is currently reactive. Say if this app can know the current location, gas level, speed and destination, and fuel mileage. With this info, the app can give the best possible gas station recommendation. It makes predictions based on data.
- Reactive application-aware applications are called studious while proactive ones are smart. Smartness definition keeps changing.

When do we adapt

- Transcoding proxy example. High-resolution images are not needed in smartphones. Transcoding proxy can deliver relevant resolution image as per requesting device. Facebook-like applications often use this. But when do we need to transcode?
- RTT: time taken by additional control messages
- Td(N): Time for download in case of no transcoding
- T(S, S'): Time taken in transcoding



- When to transcode? When Td(T) < Td(N) (based on this we can find the threshold for filesize, say Sf' - transcode when file size is bigger than this)

Module 2: Context-Aware Computing

Context Models and Context-Aware Applications

What is context

- Context is state of environment but this definition not good for mathematical analysis required for application development.
- We want to use context to adapt application's behaviour based on context.
- Why use context:
  - Context enables proactivity (samrt applications as discussed earlier)
  - To avoid interrupt[tion to user
- Context is a infinite dimensional space. Human factors and physical environment are just two broad classes of context information.

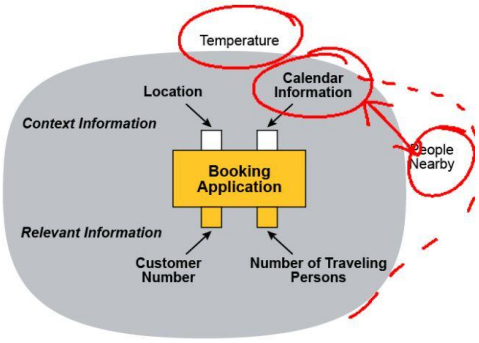
Context definitions

- Context is info that can be user by computer to enhance user experience. Eg Train booking application
- Definition by enumeration: used in research - collect whatever relevant or irrelevant info we can get in concerned scenario (its close to infinite dimension). Eg: temperature is irrelevant to train ticket but can be collected in enumeration. Its not useful but helpful to formulate things.
- Definition by relevance: Only relevant information

Definition by Relevance

Most prominent definition by Dey et al. (2001):

- *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."*



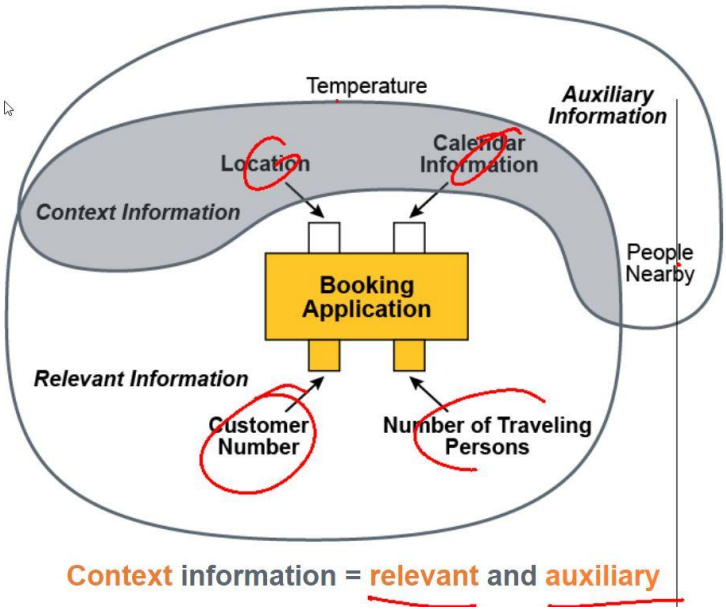
- Definition by functionality and relevance: We might even need info like number of people on station. It could be used for different application though. IOTs enable this. Using this definition we include "People nearby" in our context - earlier it was excluded.

Definition by Functionality and Relevance

Context characterizes the actual situation in which the application is used. This situation is determined by information which distinguishes the actual usage from others; in particular, characteristics of the user (location, task at hand, and so on) and interfering physical or virtual objects (noise level, nearby resources etc).

Thereby, we only refer to information as context that can actually be processed by an application (relevant information), but that is not mandatory for its normal functionality (auxiliary information).

Definition by Functionality and Relevance



Types of Context-Aware Applications

- Features of context aware applications:
  - Presentation of information and services to a user
  - Automatic execution of a service for a user
  - Tagging of context to information to support later retrieval
  - Adaptation of application's behavior and appearance
- Features: Presentation
  - Present information to the user relevant to the current situation
  - Only refers to WHICH information is presented (not HOW → Adaptation)
  - Example: Google maps showing relevant nearby places also
- Features: Execution
  - If context changes according to condition in IF-THEN rules services are automatically executed
  - Example: Auto brightness control

- Features: Tagging
  - Associating contextual information to data in order to improve later retrieval
  - Can be performed automatically or initiated by the user
  - Example: Geo-tagging (App can associate good burgers with tagged location if you tagged it) - you tag it , someone else retrieves that info later

- Features: Adaptation
  - Adapt behavior and how information is presented to given context
  - Example: Battery management apps (if battery level less than threshold - automatically shut down battery hungry soccer game)

Difficulties in using context

- Context information differs from traditional information sources in the following properties:
    - Context is gathered from heterogeneous sources (no uniform source)
    - Context is dynamic (changes frequnently)
    - Context is error-prone (sensorts are error prone)
- What we want from context-aware applications: Context-aware applications have to consider following factors:
    - **Scalability**: The application should be able to cope with a multitude of different sensors and users (adding location module to temperature sensor using application sholdn't need rewriting again)
    - **Robustness**: Stability and reliability of results, ability to adapt to new situations, resistance to frequent changes in the environment, to component failure, and to disturbing factors like noise (irrespective of sensor failures- results should be reliable)
- How to get scalability and robustness - no simple way, but structural approach can help.
- The design process can be defined as follows:
    - **Specification**: What context-aware behavior should be implemented? Which context is required for that purpose? (answer these two questions first)
    - **Acquisition**: Which sensors can be used to retrieve this context? (why its separate - its difficult based there are multiple ways to get info - eg we can get location using at least 3 modes having different accuracy and relaibility)
        - Context sources
    - **Delivery and Reception**: How is the context represented, managed and exchanged? (We need to conert raw data to knowledge bits that will be helpful for app development - so this point talks about knowledge retrieval)
        - Context models (need for knowledge retrieval)
        - Access mechanisms (how to query those models)
        - Context storage and management (we can't store raw data - how to store so that nothing is lost)
    - **Action**: Which actions should be taken corresponding to the captured context? (what we implement and how we implement - that's action)

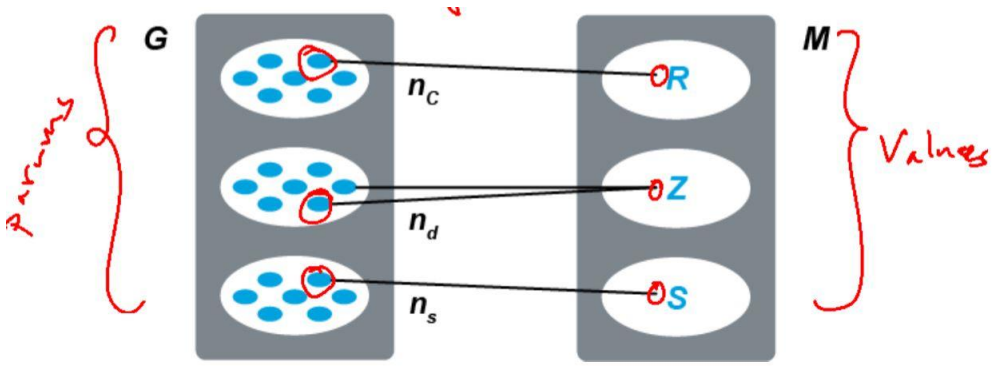
Context Sources and Models

- Coming from acquisition step above
- Sensed context
    - Query physical sensors or applications (virtual sensors)
    - Examples: temperature, outlook entries
- Inferred or derived context
    - Combining context data to gain new information ("higher level context")
    - Examples: Activity (e.g. "being in a meeting" deciding based on location and time of day), symbolic location (e.g. "S202 | A124")
- Context models - purpose is to take raw data and conert it into knowledge base
    - Context data must be represented in a machine readable form to enable an application to use it
    - Context models define the exchange of context information

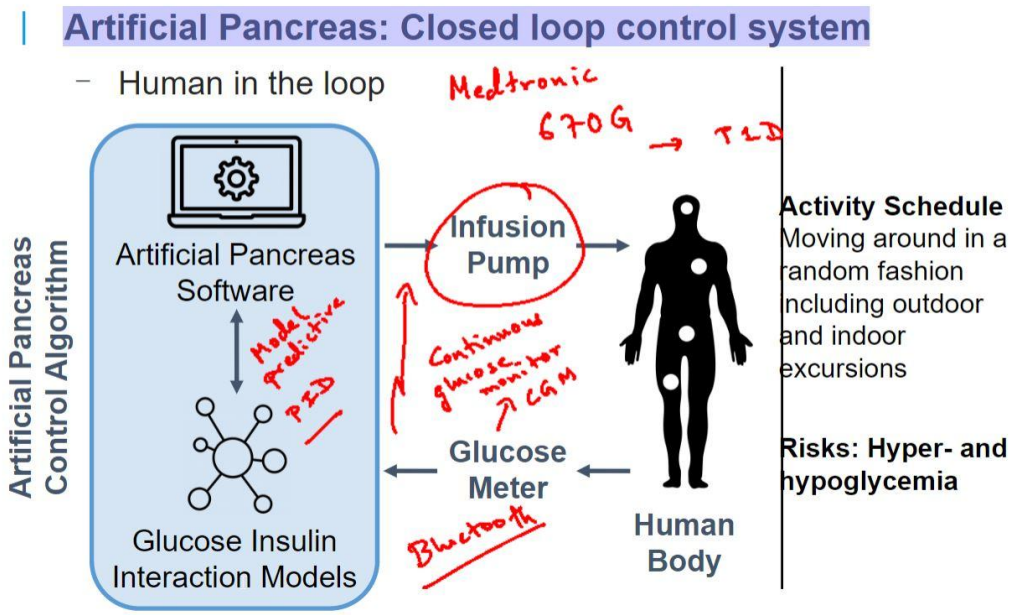
    - A context model has to provide a useful set of attributes for each context data (type, value, timestamp, source, …) – ideally, it addresses how to cope with the incompleteness and ambiguity of context information
- Various types of context models over time: Existing context models can be classified by means of the data structure they use for exchanging context information: (PPT details missing)
    - Key-Value Model (listing key value info in database format, model queried using key)
    - Markup Scheme Model (tells hierarchical structure relationship about context variables)
    - Object-Oriented Model (same as Java - typical OOPs concepts)
    - Logic-Based Model (everything is a function. Eg: locatedAt is a function that can be used to get location based on given params)
    - Ontology-Based Model (contains different concepts (say location), properties (say room number and student id) and relations between them. Based on these we get axioms.)
    - ContextFSM (Finite-State Machine) (new one)

ContextFSM

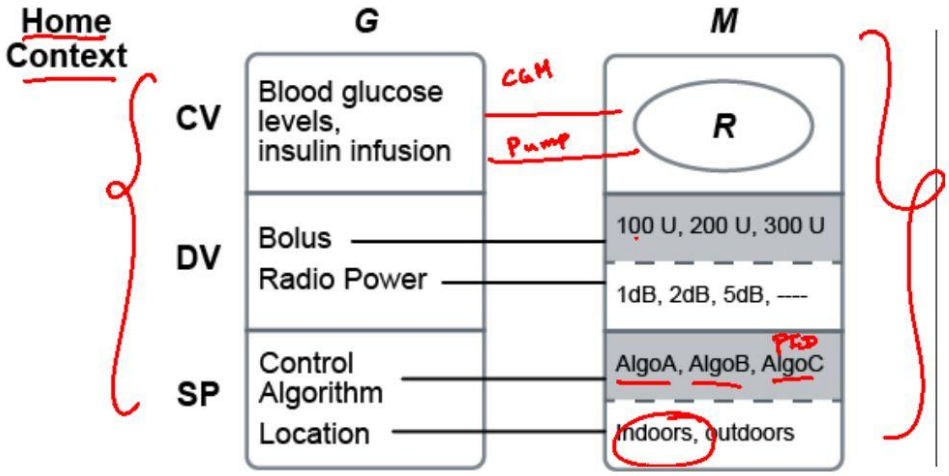
- Its a bit complex model that can be expressed in mathemataical terms.
- Context state requires state variables: System is described as a set of continuous and discrete variables
    - nc continuous variables, $vi \in R$
    - nd discrete variables, $di \in kZ$ for some $0 < k < 1$ (any natural number)
    - ns set of system parameters, $pi \in$ S , set of strings
- Context is a specific set of valuations of the system variables
    - Bipartite mapping between the variable/parameter set and the valuation set



- Using definition on an example: Artificial Pancreas: Closed loop blood glucose control system
    - Earlier type 1 diabetic subjects (no insulin in body) had to guess amount of insulin to take.
    - Different contollers exist which measure glucose level and automatically infuse insulin. Problem is - a lot of this depends on activity schedule. Activity level high means small insulin is enough to fight high glucose level. Not active means more insulin needed. So controller should adapt to prevent risk of hypoglycemia. So we need to anlayse several such context.



- Context variables in above example
  - Continuous variables (CV):
    - Blood glucose levels, insulin infusion
  - Discrete variables (DV):
    - Bolus request levels (extra insulin required based on activity. Its pre-speficified quantities only), wireless radio power levels (reception quality is dependent on radio power level)
  - System parameters (SP):
    - Control algorithms, location of user (indoors, outdoors)

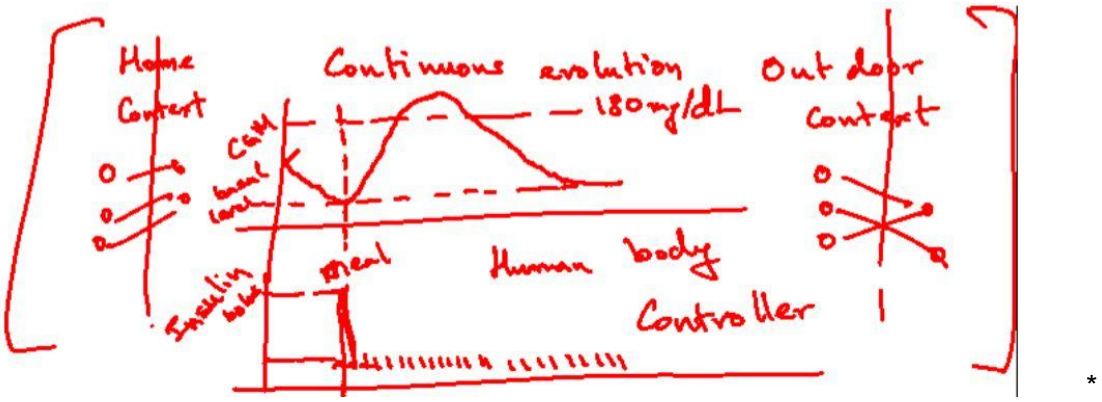


Context Analysis

- How we can use context model to anlayse context
- **What do we mean by context analysis?**
  - How does the system evolve as context changes?
    - Context change means changes in the bipartitie mapping
  - In a given mapping, the system evolution can be governed by dynamics such as differential equations (eg: SYstem evolves continuously between home and outdoor context. Plot shows

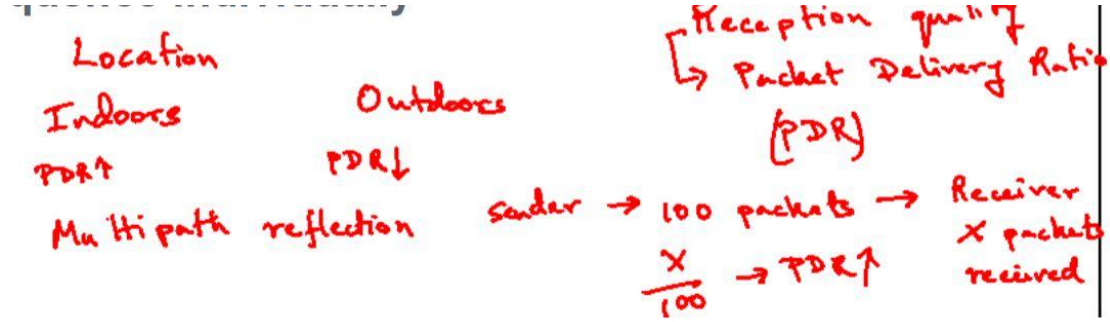
glucose level over time. Contoller adjusts infusion rate based on this evolution. So its a continuous variable.)

- As the mappings change the dynamics also change (we want to capture this continuous evolution - based on that we can evaluate if that controller is good or not)

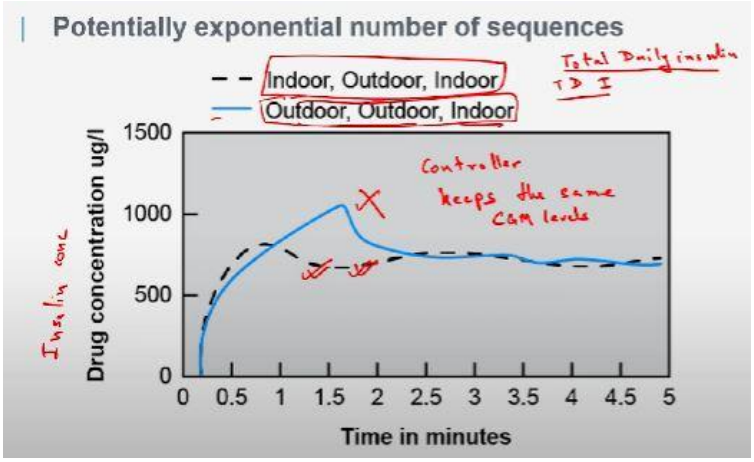


*

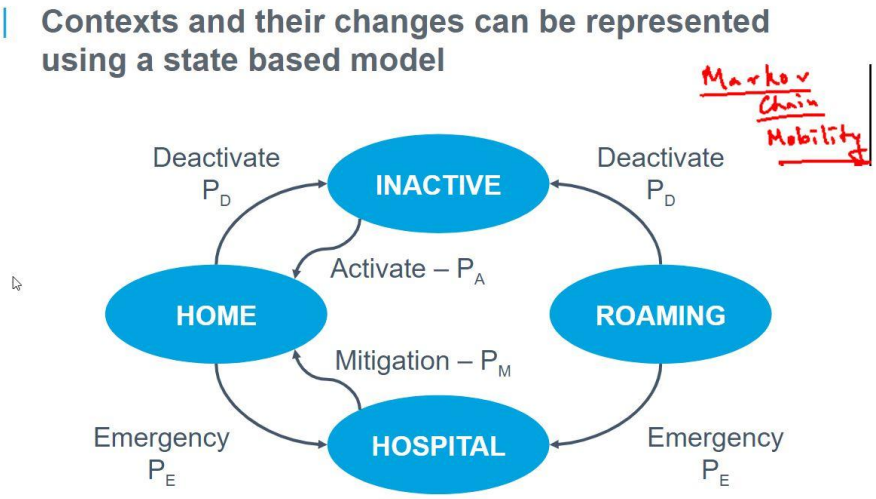
- Challenges to context analysis**:**
  - Capturing continuous evolution isn't easy. Complex differential equations
  - Environmental changes due to context have long term effects on system dynamics
  - Often the effects of context change are not memory-less
  - Hence we have to analyze each context change sequence individually
  - For example: CGM sensor values might sometimes not reach controller because of multiple factors like low Reception Quality. It depends on indoor vs outdoor setting.



- We don't have to analyse just context changes but also the **infinite possible number of sequences**. Eg: in following case we can the dotted one (because we want to sustain with minimum viable insulin)



- Context change sequence impacts controller decisions. We need to so randomized analysis (approximate modelling) to analyse these infinitely many sequences. For example: we can use Markov chain representation of mobility. We can use this to figure out most common sequences.



Context middleware

- There should be separation of concerns. So we need middleware.
  We need ways of accessing contex:
- Two ways of getting informed of context data:
  - Queries: Request context information
  - Event Subscription: The actual applications are notified every time a specified event occurs
- Consider privacy and security concerns, for example by:
  - Specifying domain dependent policy rules for access control
  - Allowing the user to control the access to his context data
    - Example: LLC (Localized Location Computation): Entity computes his location on his own

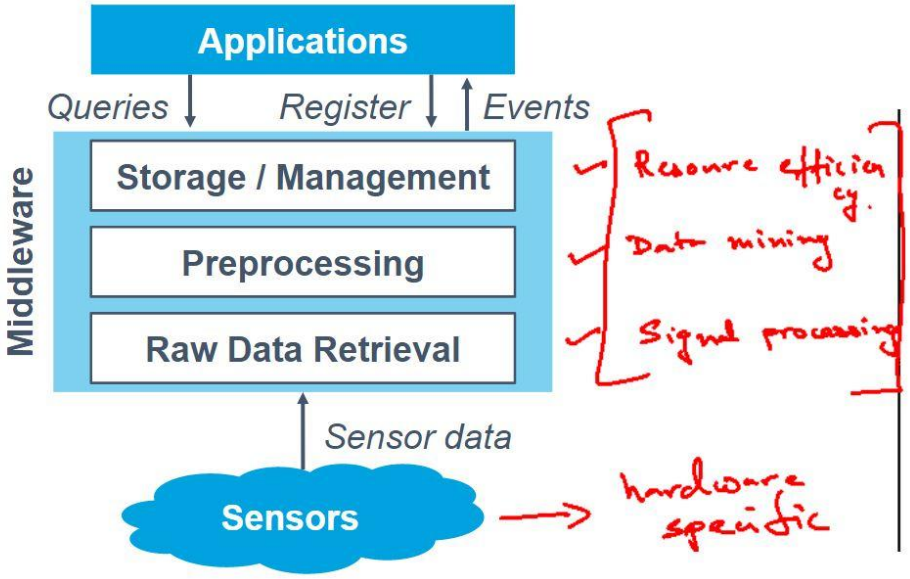
We also have to deal with context storage and management:

- Context storage and management
  - Specify a well-defined interface for accessing the context data
  - Answer queries and notify the actual applications of context changes
  - Maintain a context history or at least a context buffer
  - Provide a discovery services for the various context sources

- So we see there are many aspects involved in context-aware applications development. If programmer had to mangae all of them, it will be difficult. So there are many context management models available.
- Context Management Models
  - Widget
  - Networked Services
  - Blackboard Model
- But still even these models have to be implemented by programmers. That's where context middlewares come into picture.

Context Middleware:

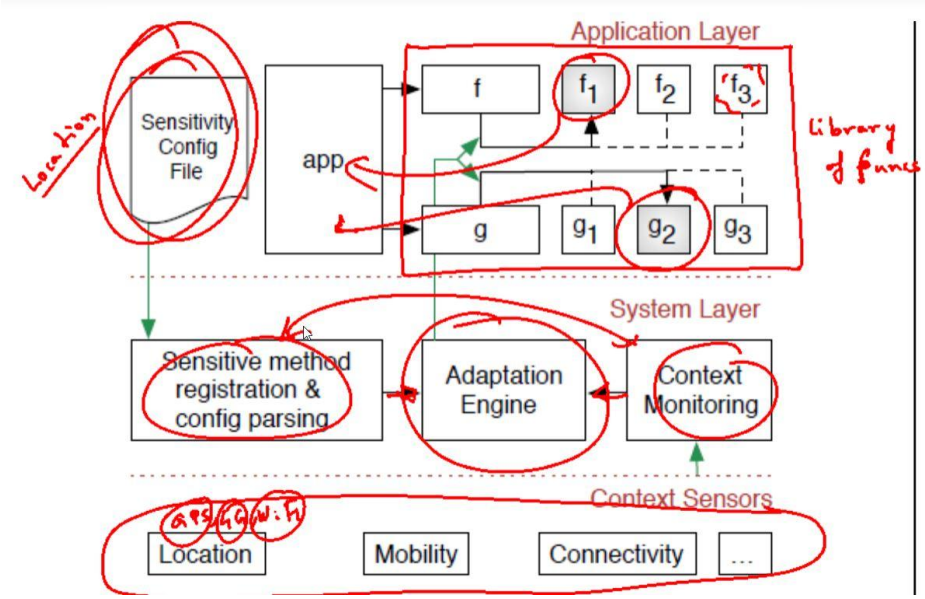
- They facilitate the development of context-aware applications by separating the detection and usage of context data
    - Use a reusable and extensible middleware for the detection
- Most middleware approaches use an architecture with the following layers



CAreDroid: Example of Middleware

- CAreDroid is a middleware for Android that makes context-aware applications
    - Easier to develop and more efficient by:
        - Decoupling functionality
        - Mapping and monitoring
        - Integrating context adaptation into the runtime
- Developers develop context-aware applications without having to directly deal with:
    - Context monitoring and context adaptation in the application code
    - **Middleware takes care of acquisition and delivery and reception**
- Developers can focus on the application logic (speciffication and action)
- At run time, CAreDroid monitors the context of the physical environment
    - Intercepts calls to sensitive methods
    - Activates only the blocks of code that best fit the current physical context.
- Context-aware methods
    - Application source code
    - The mapping of methods to context
    - Configuration files (generated by app developer - based on this middleware decides which sensor to monitor)
- Context-monitoring and method replacement
    - Performed by the runtime system
- Devloper jus gives sensitivity config file and rest middleware handles (eg: which senor to use). There is also a library of functions available. These functions helps us supply argos to guide actions based on different thresholds - what should happen if context changes - Adaption engine enables collablaoration between context monitoring and and config file
- Sensitivity Configuration File
    - The developer assigns a sensitivity list of context states with permissible operating ranges under those contexts such as:

        - Battery state, mobility state, location, etc.
- Sensitive method registration and config parsing
    - Understanding ranges of operation of each method
    - Integrate it into Android run-time system



- Adaptation engine
    - Method replacement happens at run-time
    - Extends the execution phase of the Android flow to automatically and transparently switch between methods.
- Context monitoring
    - Acquires the current context at runtime with less overhead than Android Java APIs
    - Supports raw contexts
        - The state of the hardware (e.g. WiFi connectivity, battery level)
    - Inferred contexts such as mobility status (e.g. walking, running)
- Results: A reduction of the significant line of code (SLOC) by more than a factor of 3x

Uncertainty with context

Dealing with uncertainty

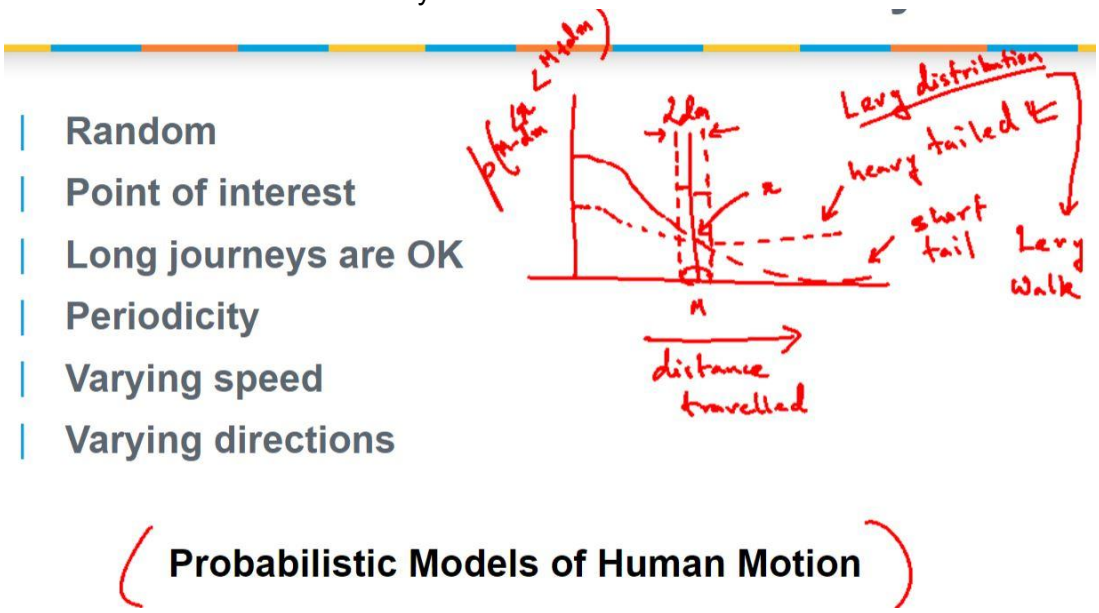
- Has to be handled in three areas:
    - Sensing context information
    - Inferring context information
    - Using context information
- How to determine uncertainty of sensed context:
    - Can be reported by sensor (e.g. biometric authentication devices give a measure for the confidence in reported data)
    - Specify a relevance function to take freshness of context data into account, because validity of context data decreases with increasing difference to the acquisition event
- How to determine uncertainty of inferred context
    - Most widely used reasoning strategies are probabilistic and fuzzy logic and Bayesian networks
- How to use uncertain context information if we are already using in application
    - Specify some qualifications to results like required confidence level (e.g. for authentication)

- Only regard the context value with maximum probability as valid

Mobility Models

Mobility models

- It is one of the major context models in MC
- Can we model how humans move?
- Why we want to do that?
    - Environment changes with mobility and so do resources.
    - Computing strategies change with the environment.
        - E.g. No network, point of interest, state of mind, food, etc.
- Mobility models are mathematical forms that tell us where, when and by how much a person is likely to move.
- Characteristics of human mobility:



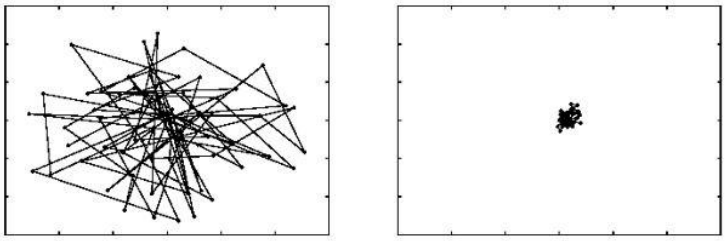
Probabilistic Models of Human Motion

- Random - first proposed model - does this first human mobility distribution? No. Humans have interests so it can't be random. But human mobility isn't fully deterministic either.
- X is variable telling distance travelled. M is particular distance travelled. Probality that distance travelled by person is in vicinity of M - doesn't follow gaussian curve (gausian means short tail). Instead probability distribution is heavy tail (that means if journey is long - its not that we are less likey to take it). So essence is **Probablity distribution of distance travelled by human can't be modeeled using Gausian curve**. Heavy tailed distribution is also called Levy distribution (human walk called levy walk)
- Periodicity exists with varying speed and directions.

Random Models of Mobility

Random Walk Model

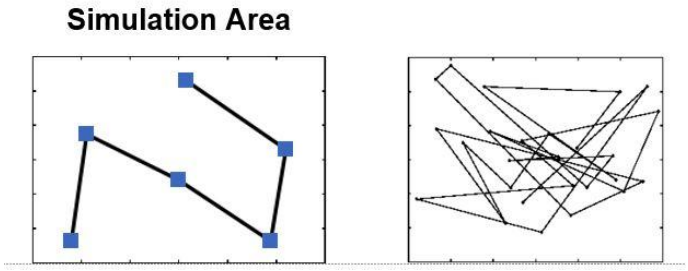
- Doesn't represent human mobility
- The Node Chooses a Random direction from [0, 2π] and a random speed from [maxSpeed, minSpeed] and moves along that direction.
- The Nodes can either move for a constant distance or for time.
- The Node repeats the above steps till the end of simulation.



- There is no provision for rest here. But humans rest.

Random Waypoint Model

- Node is initialized at a Random Location.
- The node after waiting for an initial pause time, picks a random destination and starts moving towards the destination at uniformly distributed random speed.
- Upon reaching the destination the node pauses for a Uniformly distributed random time.



- Not a perfect model - buecause humans move by interest. Here destination is random.

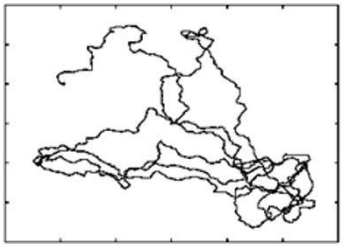
Gauss-Markov model

- More sophisticated - used markov properties but assumes step sizes and distribution are gaussian distribution - but we saw they are not true. Advantage is it enables us to hav mathematical properties

The speed and direction of the n^{th} step depends on the $(n-1)^{st}$ step:

$$S_n = \text{alpha} * S_{n-1} + [(1-\text{alpha}) * S_{mean} + \text{sqrt}(1-\text{alpha}^2) S_{xn-1}]$$

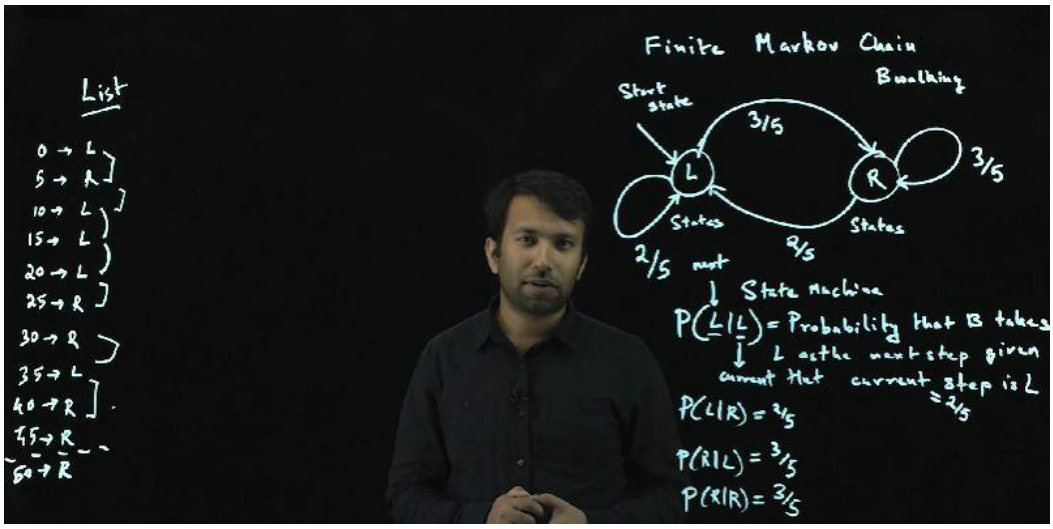
$$d_n = \text{alpha} * d_{n-1} + (1-\text{alpha}) * d_{mean} + \text{sqrt}(1-\text{alpha}^2) d_{xn-1}$$



- It utilizes history. Sn is current speed, dn is current direction. Current speed depends on previous speed sn-1 and a randomly/gaussian distributed speed. Alpha is weighing factor. Same thing for distance.
- Alpha → tuning parameter
  - Smean and dmean are mean values of speed and direction
  - Sxn-1 and dxn-1 are Random Variables from Gaussian distribution
- When alpha = 0
  - we get totally random node movement
- When alpha = 1
  - we get a perfectly linear node movement

Finite Markov Chains

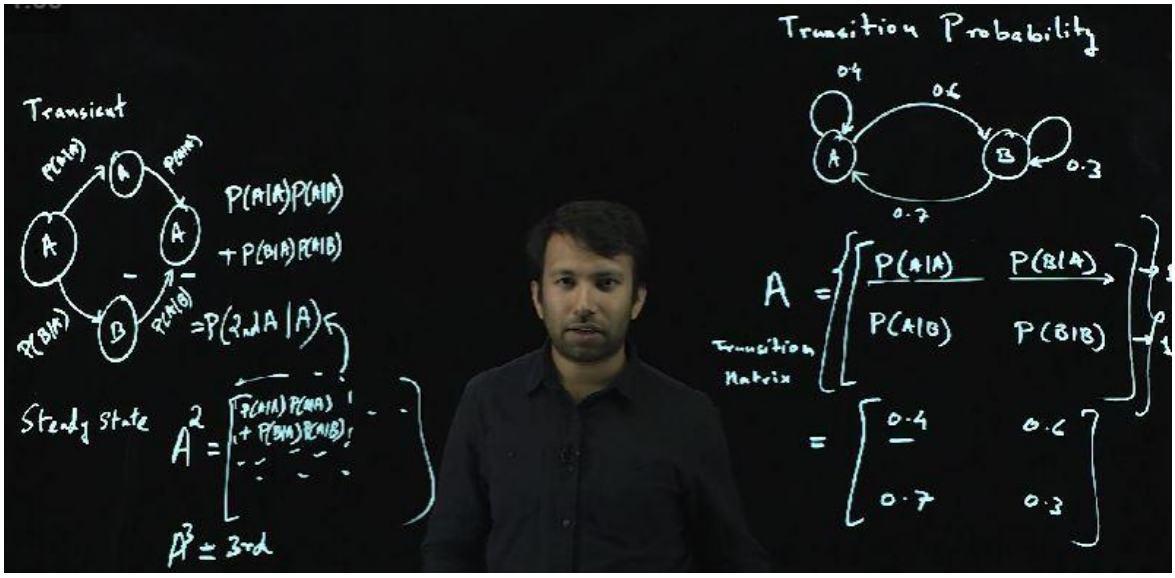
- It is one of the tools we can use to model human mobility
- Assume person A watching person B walking towards him. A is trying to develop predictive model predicting what steps B will take. Say he lists B's steps and tries to predict Left or Right step at time 't'.
- 'A' draws two circles - two states. He counts how times B was in L and took L itself as next step.A deduces that probability that B will take a L step given he is already in L is ⅗. Similarly p(L->R) = ⅗.
- The developed model is called state machine - this is Markov representation. We can write probabilities.



- Now A can predict B's next step.

Transition Probability

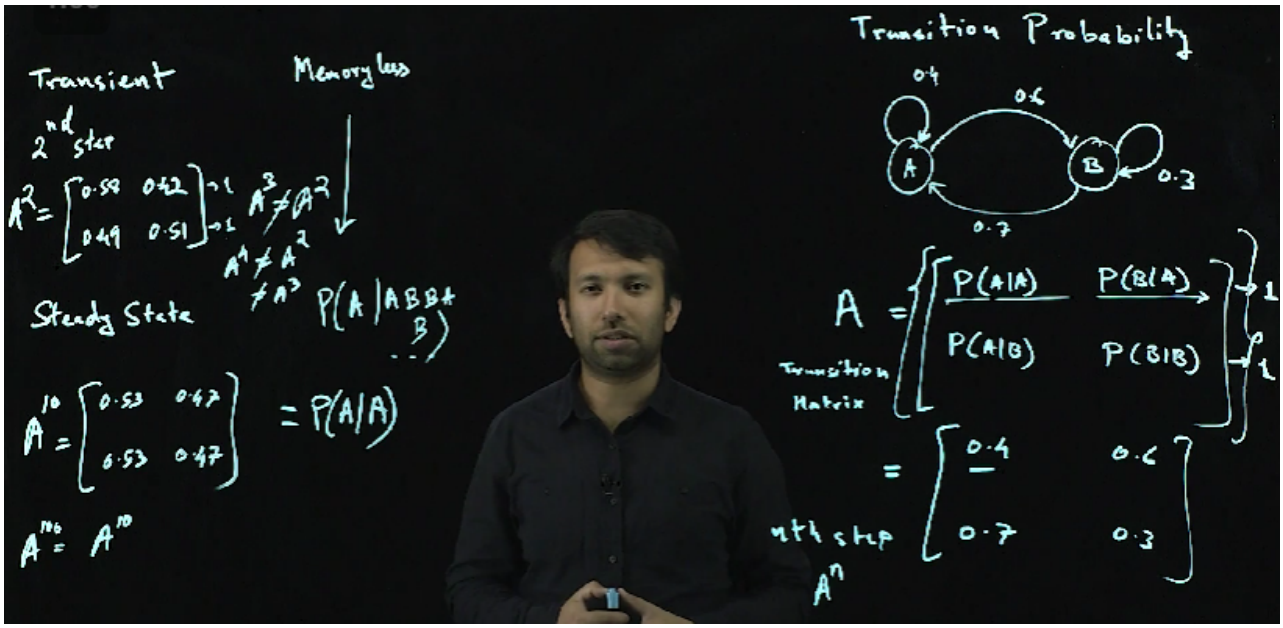
- Probalbities calculated earlier are calle transition properties.
- We can also have a transition probability matrix. Here sum of rows is 1. (column sum need not be 1)



- What is probability of next state being A given current state is A. We also want to predict future. What is probability of next to next state? This way we want to predict 10th state from now. We can get that answer from derived matrix.
    - Say we want to predict 2nd step is A given that current step is A: We will simply have p(A->A->A) + p(A->B->A) = p(A/A)*P(A/A) + p(B/A)*p(A/B)
    - Interesting thing is if we do A^2 that is multiplying transition matrix by itself - its top left element gives us 2nd step = same as one obtained above.
    - Same way if we do A^3 => we get 3rd step, similarly nth step. Given current state is A, what is probability that nth step will be A.

Transient and Steady state

- Intersting observation with Markov chain is that after certain point the transition probability matrices don't change. Say they change till A^10. But A^11~A^10 and everything after A^10 is same as A^10. When probability matrices change, they are called **transient probability matrices** and when they don't change any further => they are called **steady state probability matrices**.
- Efter seedy state is reached, we don't need to observe. So after certain point we don't need more sample (those samples are useless even if we have it). That is drawback of Markov chain model.



- **Memory less**: if we observed past few steps, it is useless i.e. p(A/ABBAB…) = P(A/A)
  So only current state matters for predicting future state (past states don't matter), so Markov chain is also called memory less.

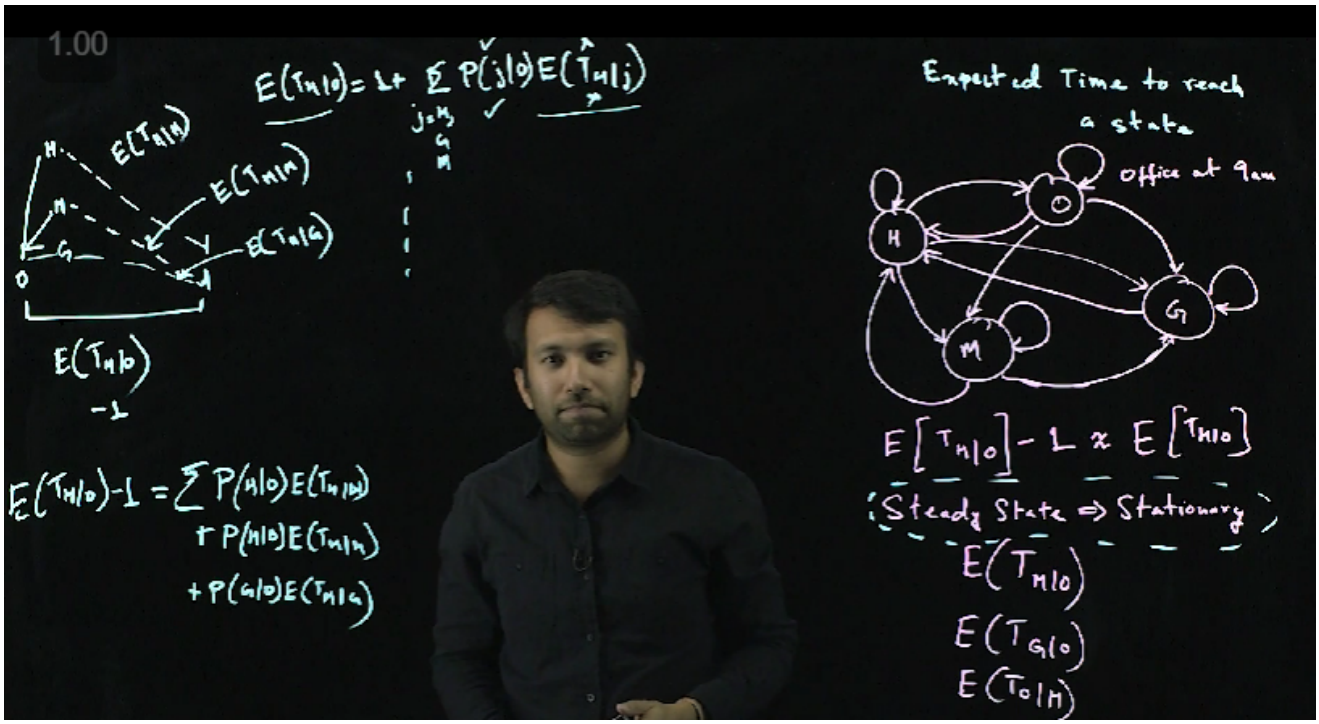
Expected time to reach a state

- We can use Markov chain model to predict probability of nth step given current step (A^n where A is transition matrix). But what if we are asked probability of person reaching home given currently he is in office. This can have various applications in smart home.
- We can obtain transition matrix for such multiple locations. H->home, O-> office, G->Grocery store, and M-> Movie. Say person is at office at 9 am, what we are asking is - what time is the person expected to come back at home? Markov chain can give us idea. We want to find E[Th/O] => expected time Th

person reaches home, given he is currently at office O (Remember we are only discussing discrete Markov chain - so we are asking expected number of time steps elapsed before he reaches home from office)

- If the steady state is already reached expected probability doesn't change with time. Even E[Tm/O], E[Tg/O] or E[To/O] doesn't change.
- Referring left part of pic, person wants to go from O to H (horizontal line). After 1 step, the remaining time is E[Th/O] - 1. If he hist first next step was H, then remaining expected time is E[Th/H]. If he hist first next step was M, then remaining expected time is E[Th/M]. If he hist first next step was G, then remaining expected time is E[Th/G]. And since we are dealing with steady state so all these expected times are almost equal. From basic probaility conjectures, we have:

  E[Th/O]-1 = p(H/O)*E[Th/H] + p(M/O)*E[Th/M] + p(G/O)*E[Th/G]



  In general we have: (left top)
  E[Th/O)] = 1 + sum over j { p(J/O)*E[Th/J] } , where J is any other location in markov model transition matrix apart from O
  And we can have similar equations for other 3 locations as well i.e. Office O, Movie M and Grocery store G => that gives us 4 unknowns and 4 equations. We can solve them and get expected time to reach a state (recall that we already know the probabilities, what we don't know is expected time E)
- Now we know expected time a person will reach home from office => we can do smart things.

Machine learning for context modules

Cognitive Mobile Computing (CPM)

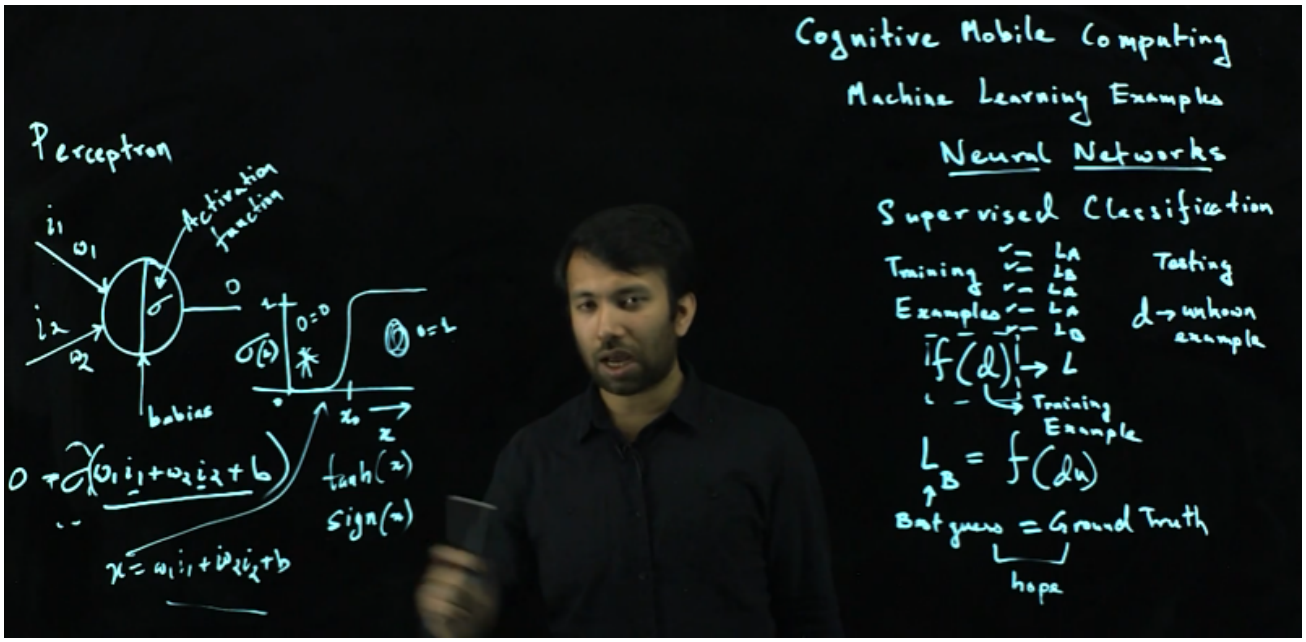
- Coginitive relates to brain. So Cognitive Mobile Computing means getting context info from user's brain.
- Brain senor enabled applications can blur movie scenes or even provide different endings to same movie.
- To extract knowledge from such sensor, we need ML systems.

CPM Challenges

- Standard architecture is sensor sends raw singal to some edge device like smart phones which forwards this to ML application in processed or raw form. ML application gathers knowledge which is passed back to edge device and then actuation is sent back to same sensor or separate actuator like some screen display.
- One problem is EEG data is fast and huge. SO it becomes classic big data application => lot of data (variety), lot of variety in data and data coming very fast (high velocity means we need fast proccessing/feedback loop). Problem is: Where to do the computation. Typical response time is 300 milliseconds, transmitting and processing so fast is difficult.
- Second challenge: Raw brain data is meaningless for humans. So developing algorithms is difficult, so ML systems which can learn from examples are our only bet. ML systems need ground truth. But with brain its difficult to have ground truth. Mental states depend on physchology of person so getting ground truth is difficult. Good ML algos are thus difficult to develop.

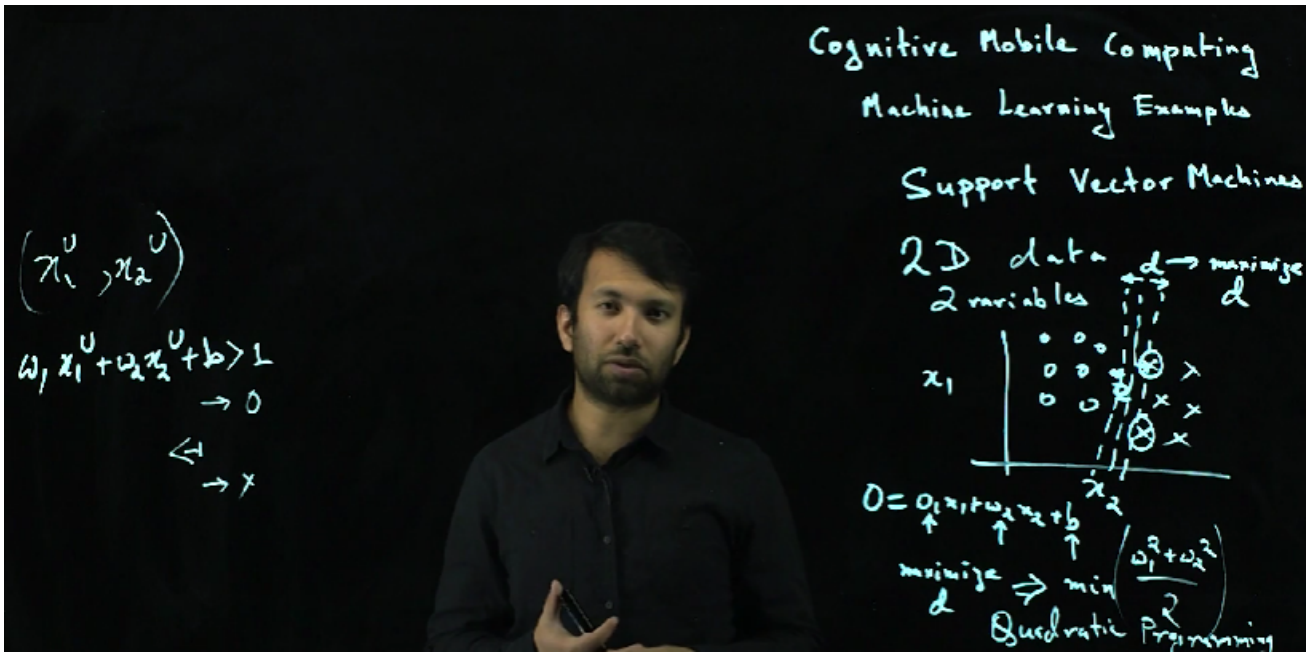
CPM: ML examples: Neural Networks

- In supervised classification, aim is to learn function f() that can predict correct label.
- Perceptron is basic Neural Network. It contains linear part (w1i1 + w2i2 + b) and then a non-linear activation function called sigma acts on this linear part (figure left) to get discrete class output. tanh(x) and sigma(x) are some of such non-linear functions.



CPM: ML examples: Support Vector Machine

- Assumes that dataset with two classes can be classified by just drawing a straight line.
- Equation of line is w1x1+w2x2+b (SVM needs to figure out w1, w2 and b)
- Inherent assumption in SVM is which line we can choose - whatever line you choose, both classes will have closes points (perpendicular distance d) to those lines - we take that line that maximises the distance d. Maximizing d implies minimization of (w1^2 + w2^2) / 2

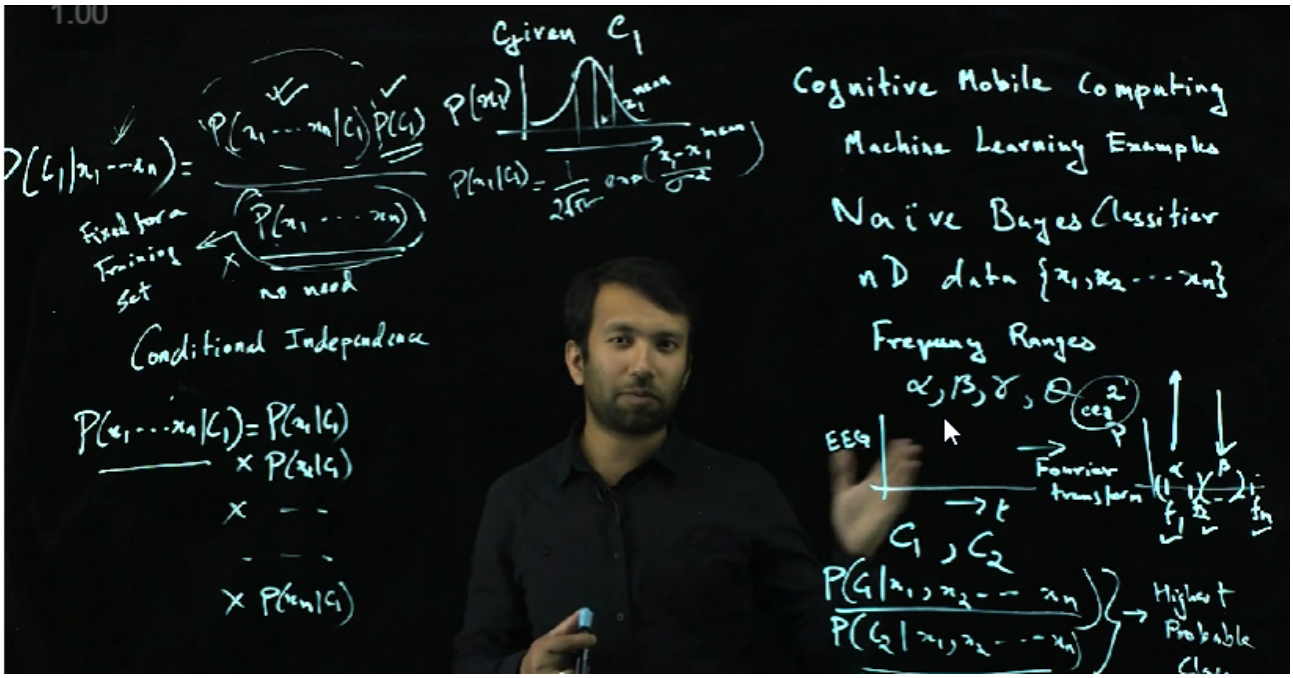


- After training we can use calculated w1 and w2 on unknowns (x1u, x2u) inputs and if w1x1u + w2x2u + b > 1, then label is o, else label is x. That's how we test.

CPM: ML examples: Naive Bayes Classifier (NBC)

- Simpler than NN or SVM but works best with brain sensors.
- If we take fourier transform of EEG vs time series plot, we get a power distribution i.e. eeg^2 (power distribution vs frequency plot). Every time series signal can be cperressed as summation of sign waves. The frequency on x axis are the frequencies of those sign waves and magnitude of those sign waves is power (eeg^2).
- Different frequency ranges are very intersitn for brain signals. Power varies across those frequency ranges in brain signal. With those we can figure out mental states.
- Brain data is multideimesional, SVM can't handle. Simple NBC can handle multiple dimensions better. Say x1, x2, … xn are parameters and say there are two classes c1 and c2. At a particualt time we have some values of x1, x2, …., xn. What NBC does is - it tries to find out what is probability that it is say class 1 given those xi values.
  So it finds p(C1/x1,x2,...,xn) and p(C2/x1,x2,...,xn) and then outputs highest probable class
- We can now use Bayes rule:
  p(C1/x1x2…xn) = (p(x1x2….xn/C1)*p(C1))/p(x1x2...xn)
  We already know p(C1) => count number of samples with class C1 in training set divided by total samples
  Denominator p(x1x2...xn) is difficult to obtain but this value is fixed for a traning set. So we don't need to compute it. Only thing remainig is p(x1x2….xn/C1). NBC uses simplistic assumption of conditional independence. i.e.:

p(x1x2….xn/C1) = p(x1/C1)*p(x2/C1)*.....*p(xn/C1)

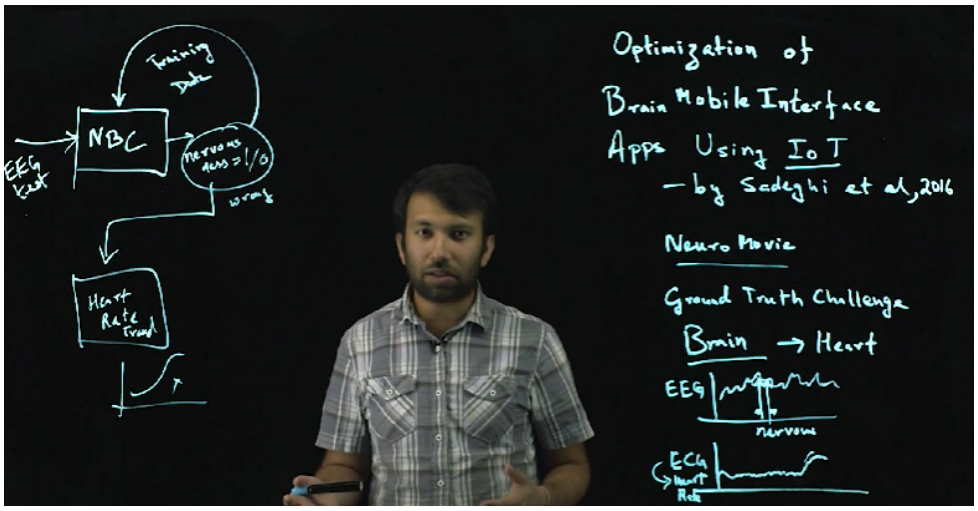


If xi is distributed in gaussian form, we can write gaussian expression for p(xi/C1)=(½ root pi) * something

So that we we can get p(C1/x1x2…xn) and similarly get p(C2/x1x2…xn)

Ground truth challenge

- For brain-mobile interfaces, read paper ”Optimization of Brain Mobile Interface apps using IOT by Sadeghi 2016”.
- This paper takes application of neuromovie and tries to identify challenges.
- First challenge was Machine learning.
- Ground truth challenge was another one.

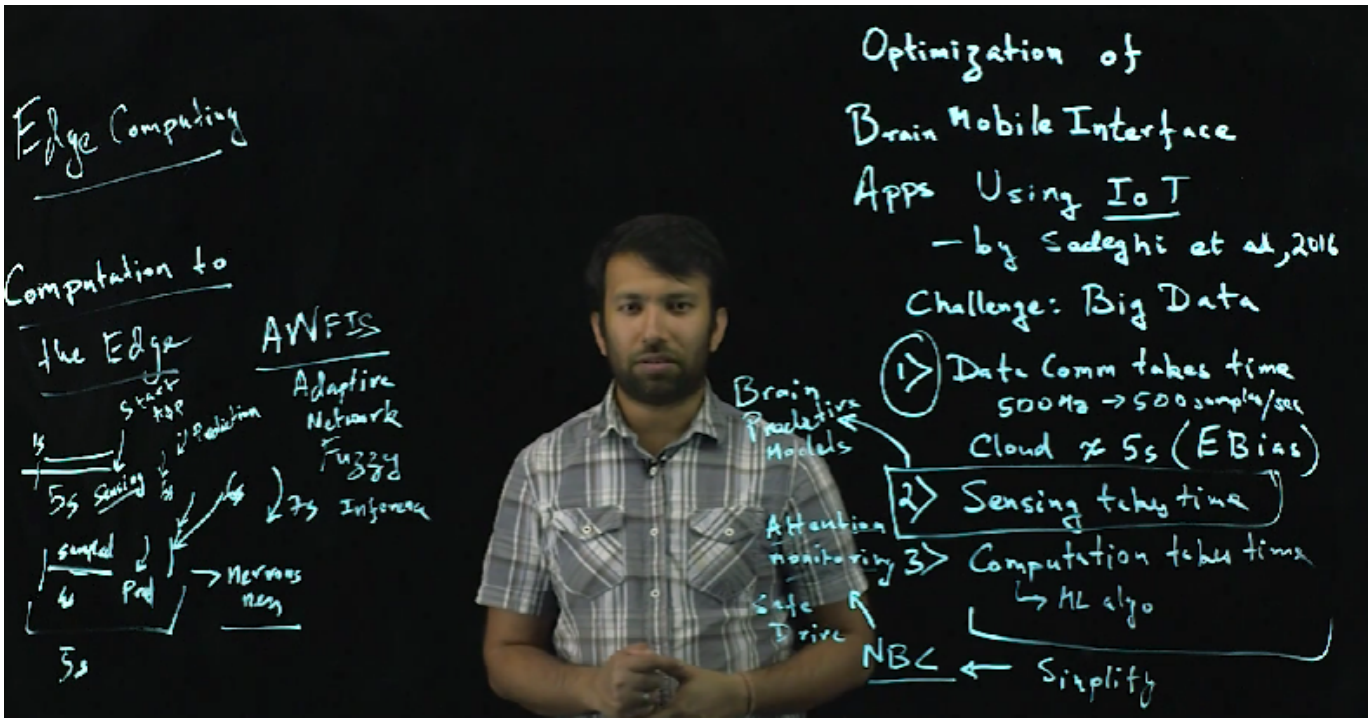


- Generally ground truth is obtained by manual labelling. We can monitor attention span or nervousness of person but now it depends on psychology of person. Same scene may not make everyone scared. One solution to this is to get secondary (surrogte) sensors. Whatever happens in brain is also experienced by heart say, so we monitor heartbeat also, heavy breathing etc. Heat may respond bit later (ECG) but we need both primary and secondary sensors to make valid conclusions and proactive

systems. We can continuously monitor ML model outputs with heart rate trends. This enables continuous revalidation and feedback based on auxiliary signals.

Big data challenge

- High amount of data coming and need to be processed within 300 ms.
- Roadblocks:
    - Data communication takes time (500 hz singal means 500 samples per second) but Ebias is around 5 sec at least (time need for cloud communication)
    - Sensing takes time
    - Computation takes time (ML algo)



- What to do: (steps told in profs paper)
    - Computation: First try to simplify computation - example use NBC
    - Data communication: Look for edge computing: Two ways to do this
        - Send data to edge, then send to cloud; cloud does computation
        - Send code from cloud to edge and edge get data from phone, compute on edge device
    - Sensing time:
        - Use brain predictive models (Say we have past EEG samples, we train on those, with that try predicting next sample) Say take 4 second sample and try predicting 5th second sample - that way every second we have new nervousness value. This prediction technique is called ANFIS (Adaptive Network Fuzzy Inference)

Advantages of Prediction

- Earlier last point we saw that we could overlap sensed data and prediction. It increases responsiveness.

- Another advantage is it can save power and energy. We have sensed data dn predicted data. Consider sensor -> edge (does some mining and prediction) ⟶mined model⟶ cloud (does same prediction as edge). So sensed data is available with sensor+edge and predicted data is available at edge+cloud. Edge is common in both sensed data and predicted data. We can take advantage of this. If edge sees that its predicted is very accurate, instead of sending mined model, it can send a different signal to cloud (say “match”) seeing which instead of again mining new model, cloud can reuse old mined model itself. Thus we can save on communication. Hence we decrease duty cycle of radio saving radio power.

BraiNet

Network of Brains

A Framework for Cognitive Mobile Computing
IMPACT Lab Project

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Motivation

- **Pervasive brain monitoring**
- **Recognizing mental states on-the-go**
- **Providing online feedback to the user**

- **Solution**
  - Managing execution of tasks in an Internet of Things (IoT) setting, enabled by BraiNet architecture.

Challenges

- Large amount of high frequency data
- Accurate recognition of mental states
- Real-time feedback to the user
- Energy efficient processing on mobile

Big Data

Ground Truth

Ground truth is not much available in psychology

data transfer to cloud is heavy

- **BraiNet Architecture**
  - Offload **and** execute in real-time constraints

Edge

Machine learning Prediction

offloading introduces to edge computing

What is BraiNet?

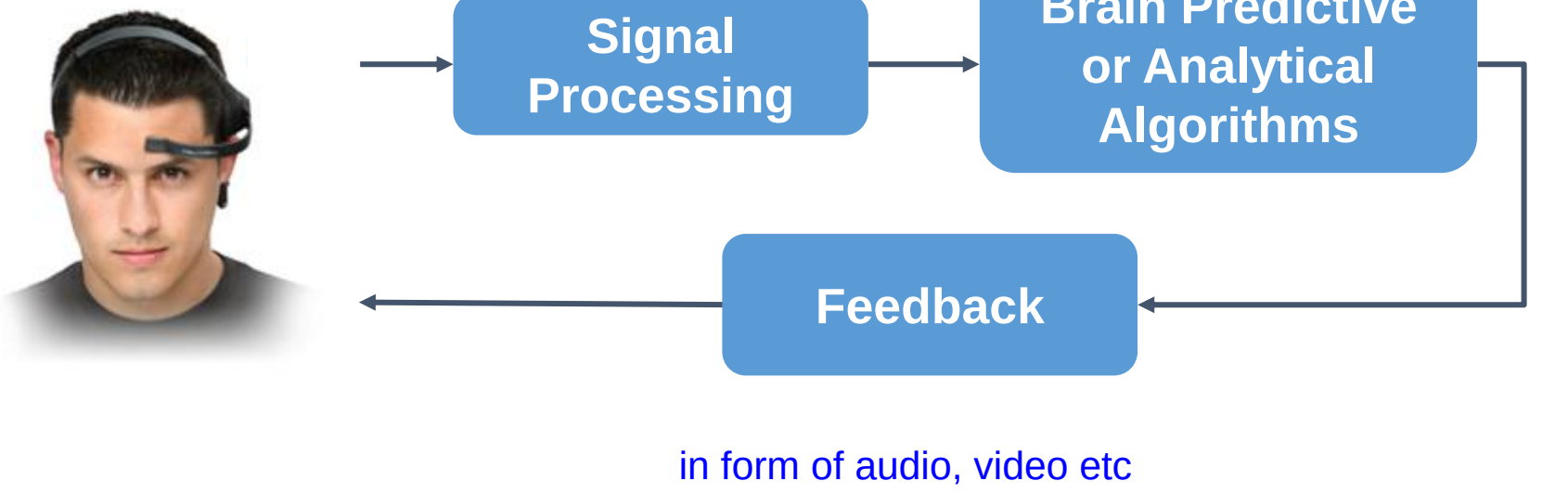
- **BraiNet is a network of individual Brain-Computer Interface (BCI) systems.** Connect brain to cloud
- **BraiNet is a pervasive brain monitoring system to develop BCI apps at a societal level.**

Telepathy

research success in France and Japan

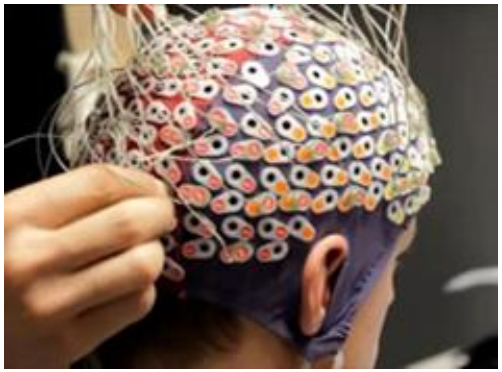
Brain-Computer Interface Systems

- **BCI systems have three (3) main phases:**
  - Data acquisition
  - Signal processing
  - Application



Data Acquisition

Usable Sensors



108 Channel Medical Grade Sensor

Neurosky Wearable Sensor

single channel

14 channels

Emotive Wearable Sensor

Modern sensors are non-invasive

we can lose some fidelity of data depending on sensor

Signal Processing

FFT plots signal power over frequency

- **Frequency Band Cognitive State**
  - Delta (0.5-4 Hz) Sleep Activity
  - Theta (4-8 Hz) Attention Level
  - Alpha (8-13 Hz) Relaxation at decreased Attention Levels
  - Beta (13-30 Hz) Active Concentration and Alertness State
  - Gamma (30-100 Hz) Perception
- **Features can be extracted from different domains on the brain signal including time and frequency.**

Based on frequencies, we can tell things like you are eyes closed but thinking hard

BraiNet

Network of Brains

Machine Learning Introduction

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Machine Learning

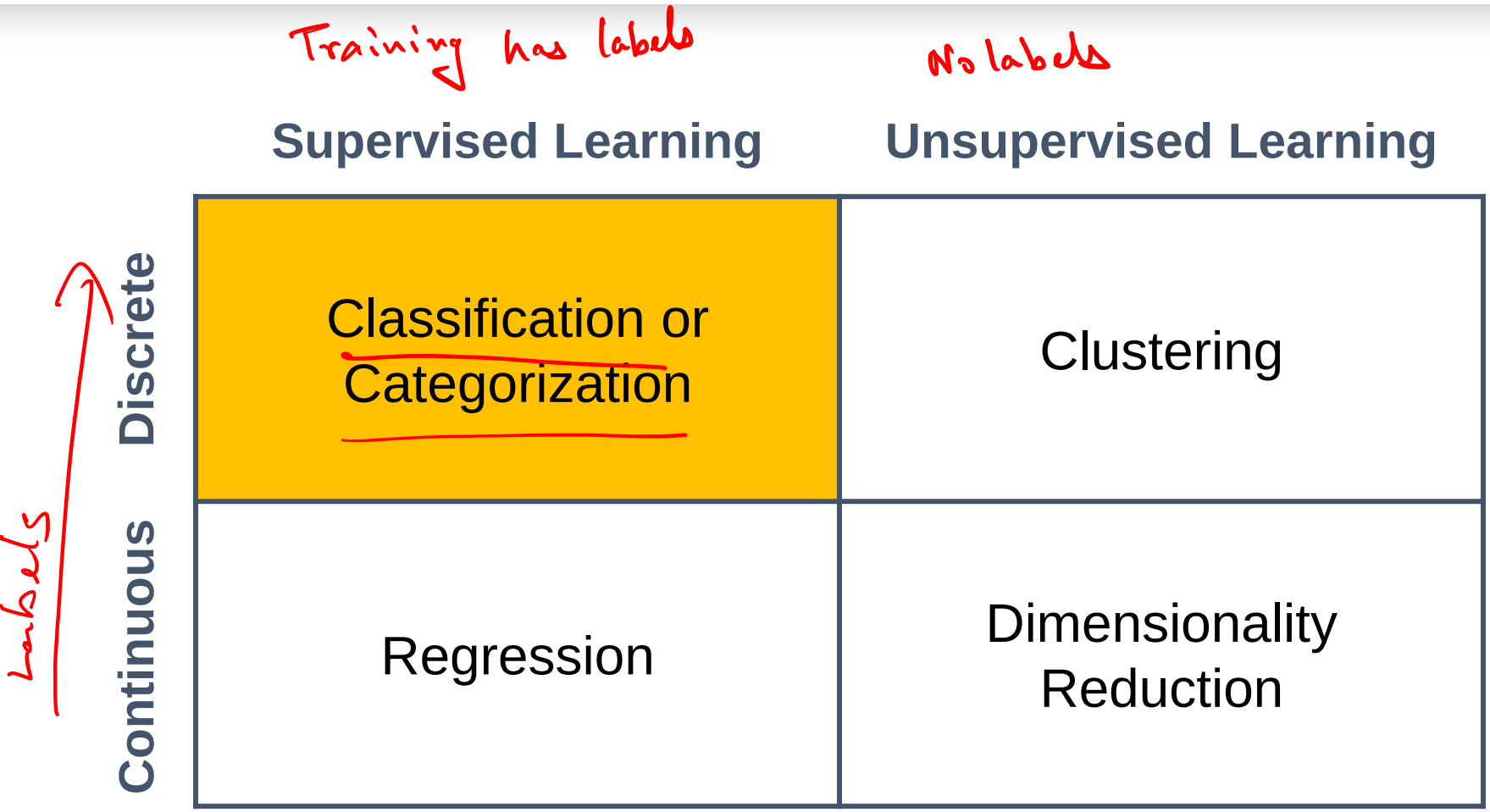
We used supervised algorithm

Given a set of data, can we define classes?

Difference between data points

classify data points in classes

Machine Learning Problems



Machine Learning Framework

- **Apply a prediction function to a feature representation of the image to get the desired output:**

f() = “apple”

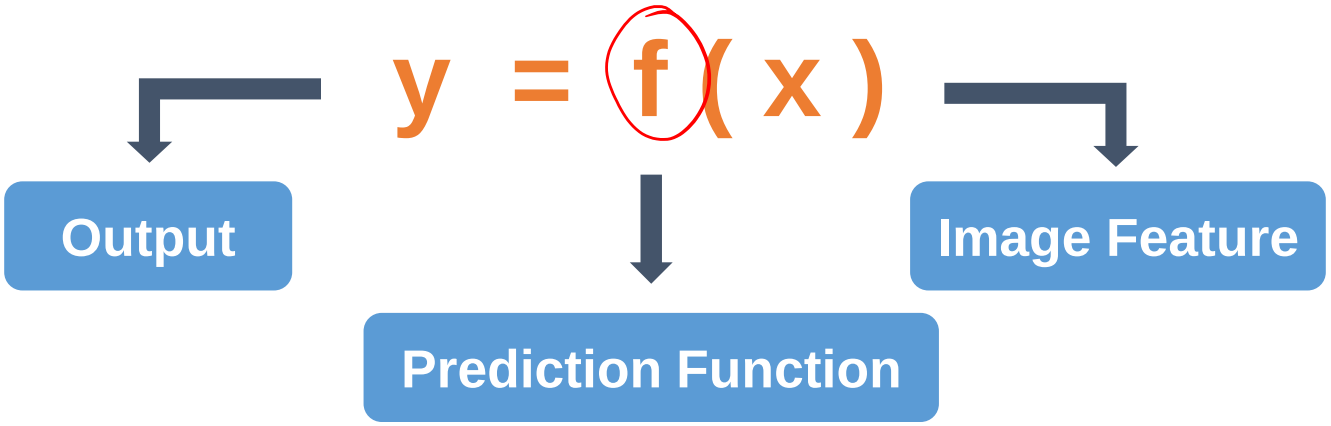
f() = “tomato”

f() = “cow”

Our goal is to learn this function f through training

Modified from: L. Lazebnik

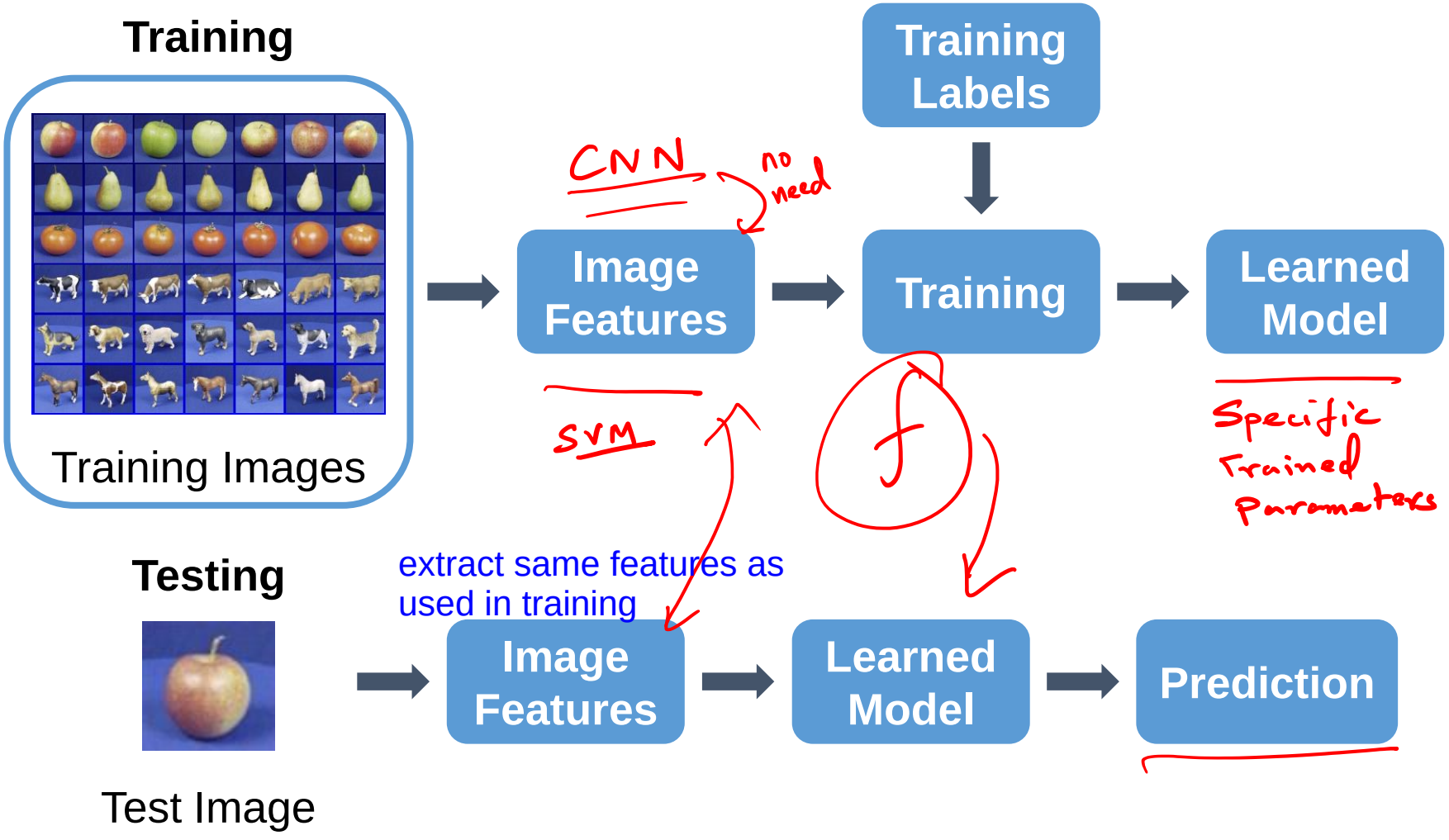
Machine Learning Framework



- Training: Given a training set of labeled examples $\{(x_1,y_1), \ldots, (x_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- Testing: Apply f to a never before seen test example x and output the predicted value $y = f(x)$

Modified from: L. Lazebnik

Machine Learning Steps

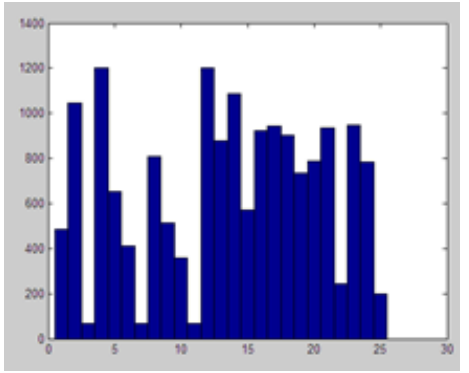


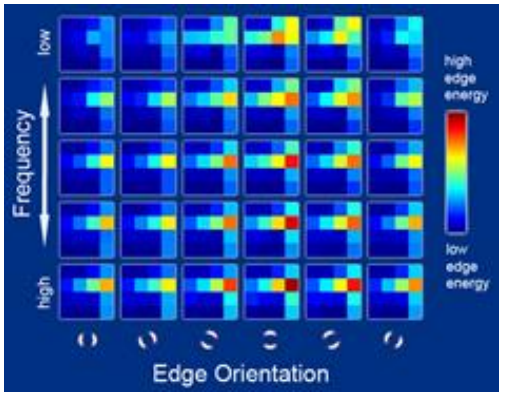
Modified from: D. Hoiem and L. Lazebnik

Machine Learning Features

Extracting features:

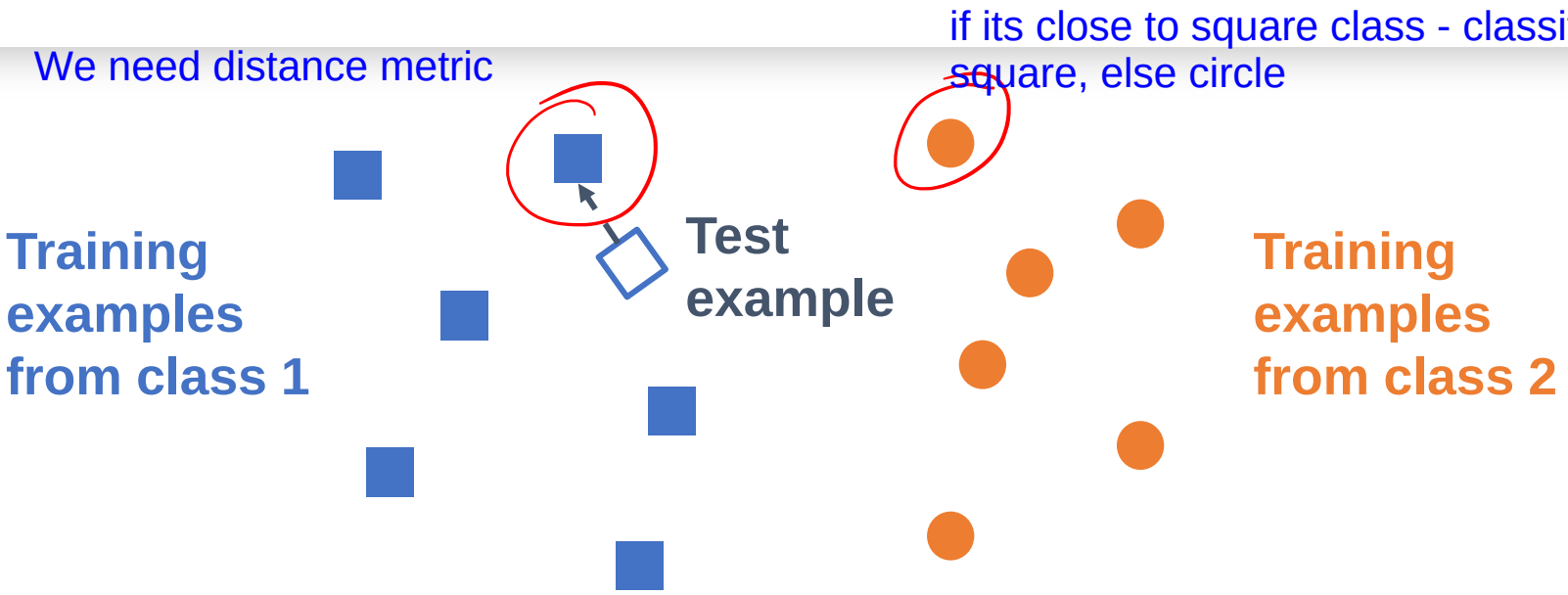
- Raw pixels
- Histograms
- GIST descriptors





Modified from: L. Lazebnik

Classifiers: Nearest Neighbor

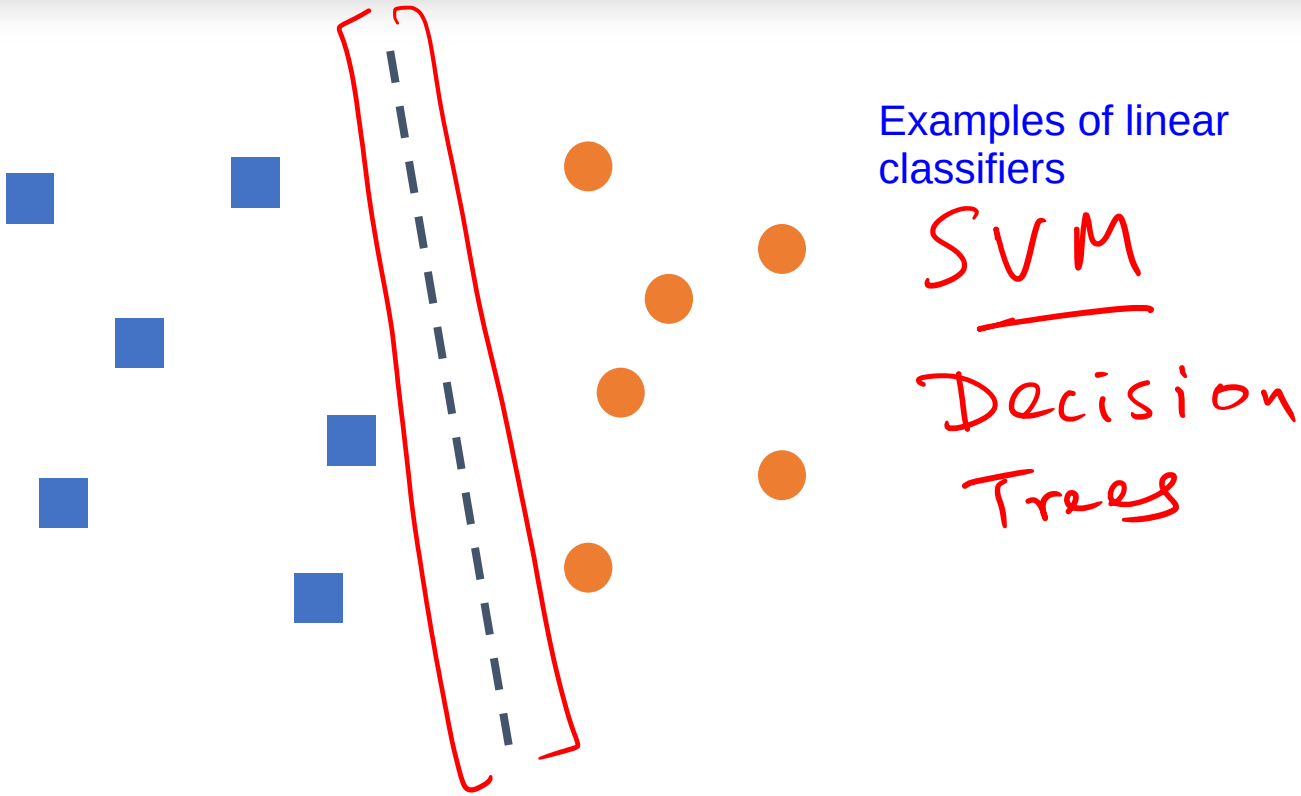


f(x) = label of the training example nearest to x

- All we need is a distance function for our inputs.
- No training required.

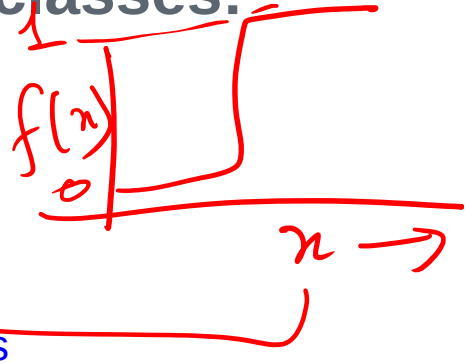
Modified from: L. Lazebnik

Classifiers: Linear



- Find a linear function to separate the classes:

$$f(x) = \text{sgn}(w \cdot x + b)$$



Modified from: L. Lazebnik

This is how we convert a linear classifier into discrete classes

Classifiers

- **There are many classifiers from which to choose:**
  - SVM
  - Neural networks
  - Naïve Bayes
  - Bayesian network
  - Logistic regression
  - Randomized Forests
  - Boosted Decision Trees
  - K-nearest neighbor
  - RBMs
  - Etc.
- **Which is the best one?** This is what we will discuss.

Modified from: D. Hoiem

BraiNet

Network of Brains

Generalization

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Generalization

Take some traits of people and see if these fits on others that is how ML also works.
If we see new trait, machine might get classification wrong

Training set (labels known)

Test set (labels unknown)

How well does a learned model generalize from the data it was trained on to a new test set?

Every machine has some error component coming because of generalization. We wan tto minimize it but can't make it zero.

Modified from: L. Lazebnik

Generalization

If we learn very simplistic and insignificant details from some set of people and then apply them on everyone else later, we will make errors. That is bias. (say we learned background noise levels - they are useless)

Components of generalization error:

- Bias: How much does the average model over all training sets differ from the true model?
- Error due to inaccurate assumptions/simplifications made by the model
- Variance: How much models estimated from different training sets differ from each other

Variance means we learned all nitty gritty details of few people and then assume that all other will also follow the same.

Model shouldn't depend on type of sensor used - that's insignificant detail

We want to balance bias and variance

Generalization

- **Underfitting: Model is too “simple” to represent all the relevant class characteristics**
  - High bias and low variance
  - High training error and high test error
- **Overfitting: Model is too “complex” and fits irrelevant characteristics (noise) in the data**
  - Low bias and high variance
  - Low training error and high test error

Modified from: L. Lazebnik

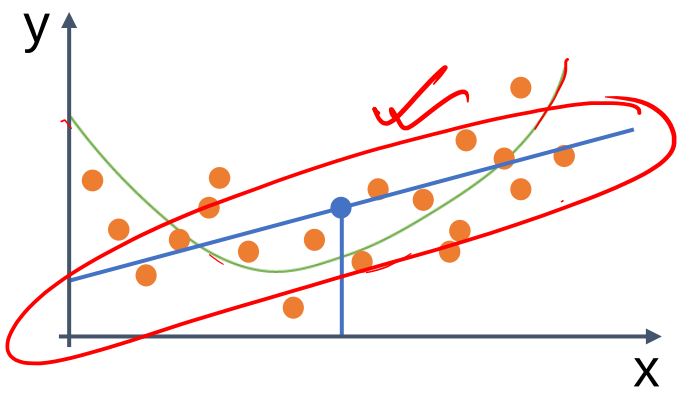
No Free Lunch Theorem

We can't minimize bias and variance same time.

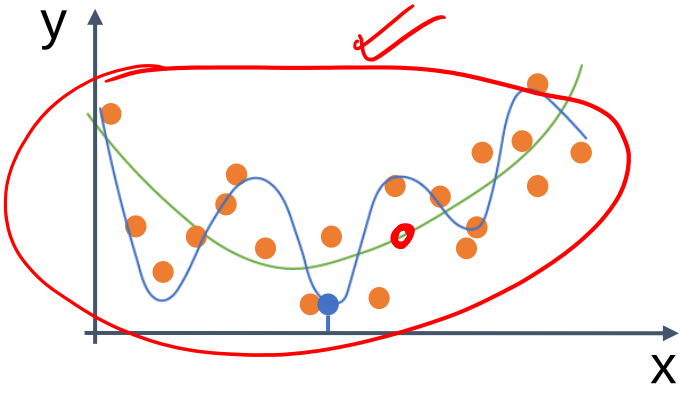
Modified from: D. Hoiem

Bias-Variance Trade-Off

If we reduce one, other one will increase



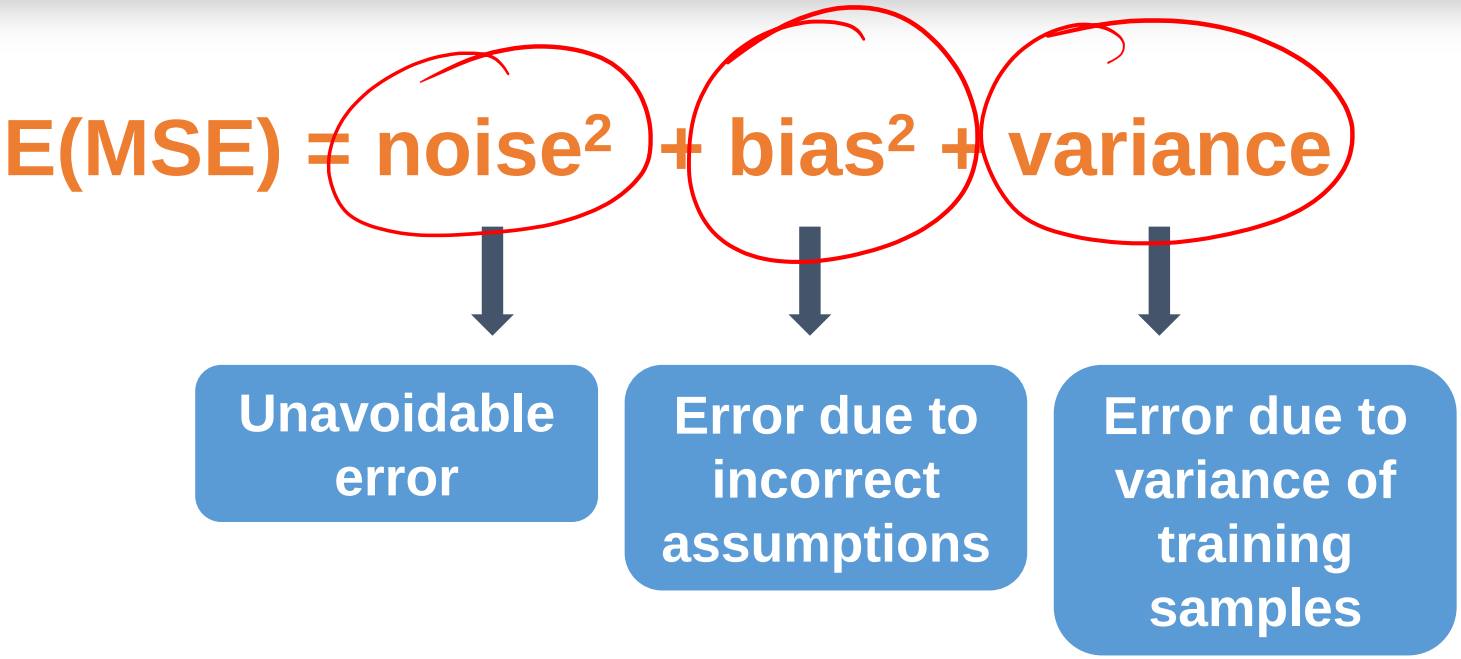
- **Models with too few parameters are inaccurate because of a large bias (not enough flexibility).**



- **Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).**

Modified from: D. Hoiem

Bias-Variance Trade-Off

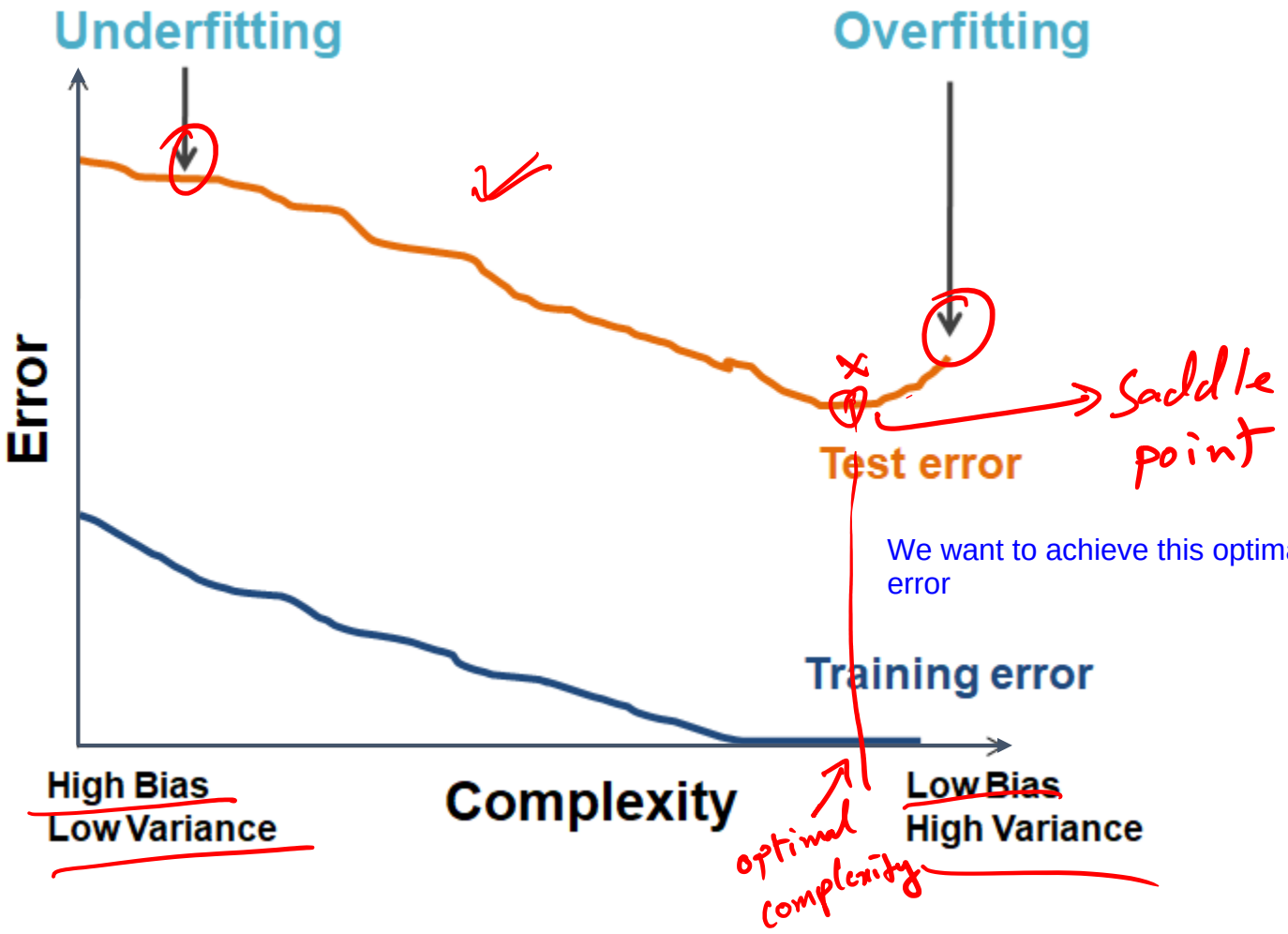


See the following for explanations of bias-variance (also Bishop's "Neural Networks" book): The Bias-Variance Tradeoff

Generalization error is sum of all these errors.

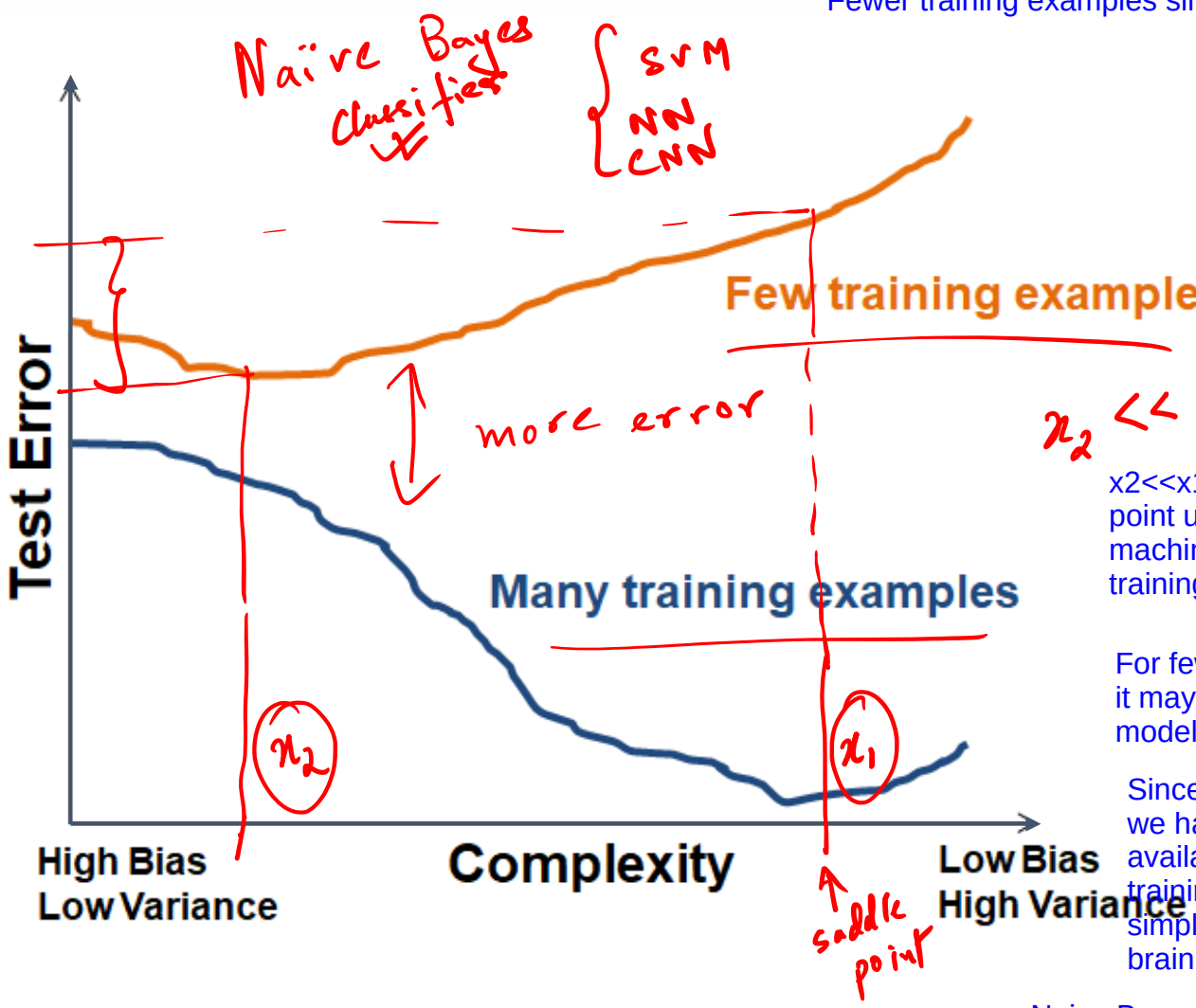
Modified from: L. Lazebnik

Bias-Variance Trade-Off



Modified from: D. Hoiem

Bias-Variance Trade-Off



Fewer training examples simply imply more error

$x_2 << x_1$

x2<<x1 means there is no point using higher complexity machines if we have very few training examples

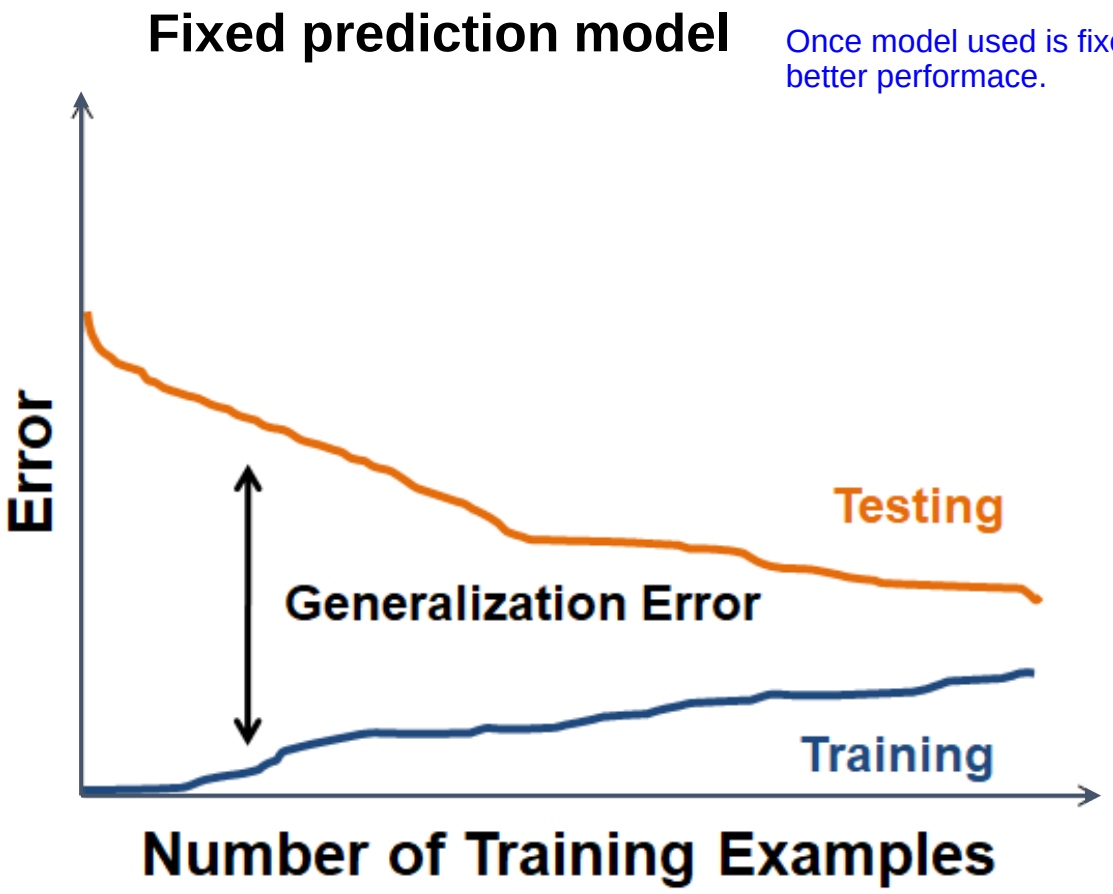
For fewer training examples - it may be wiser to use simpler models

Since in brain mobile apps, we have less ground truth available so we have fewer training examples, thus simpler models work better in brain mobile machines.

Naive Bayes work better than SVM, CNN etc in such cases.

Modified from: D. Hoiem

Effect of Training Size



Modified from: D. Hoiem

BraiNet

Network of Brains

Infrastructure and Applications

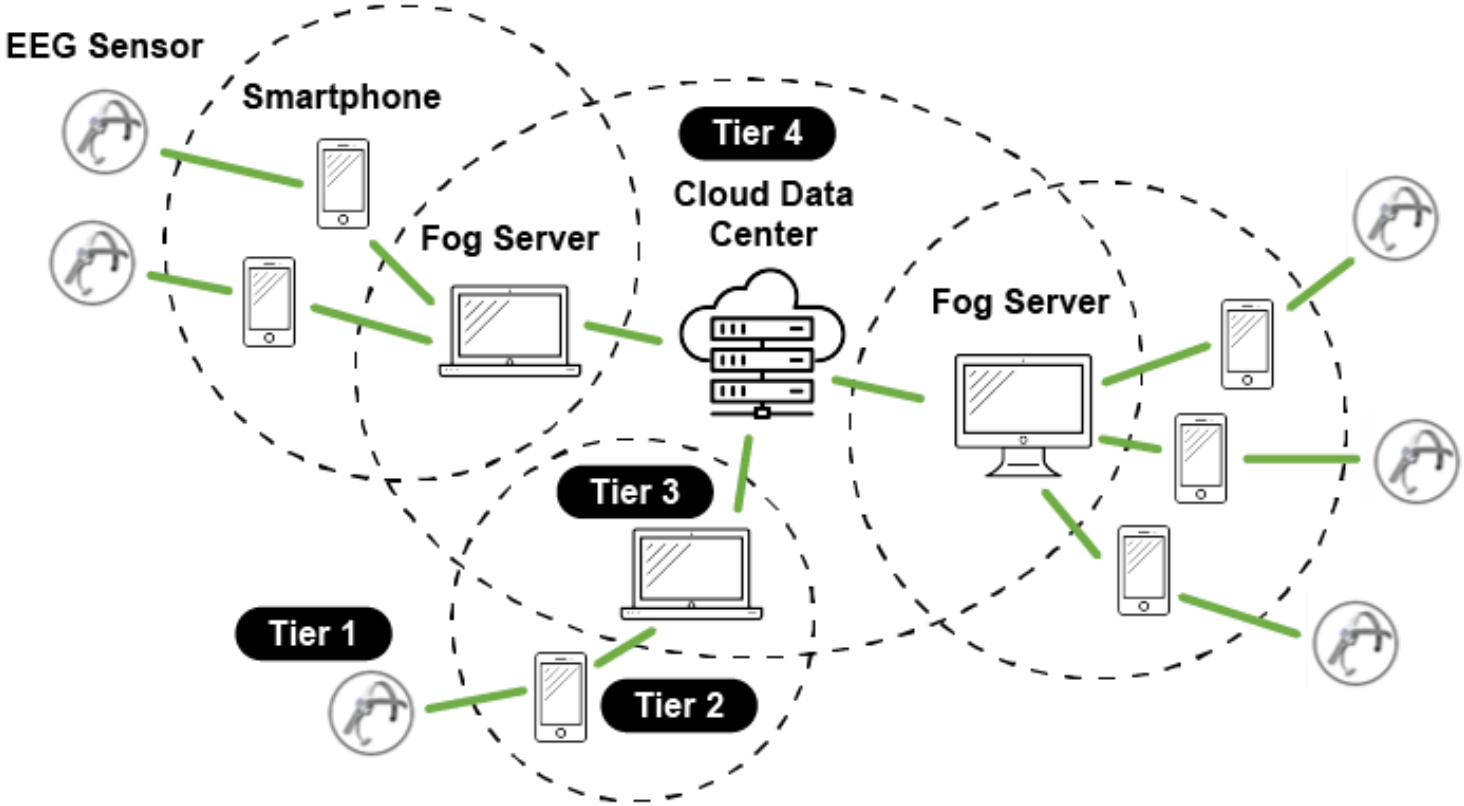
Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

BraiNet Infrastructure

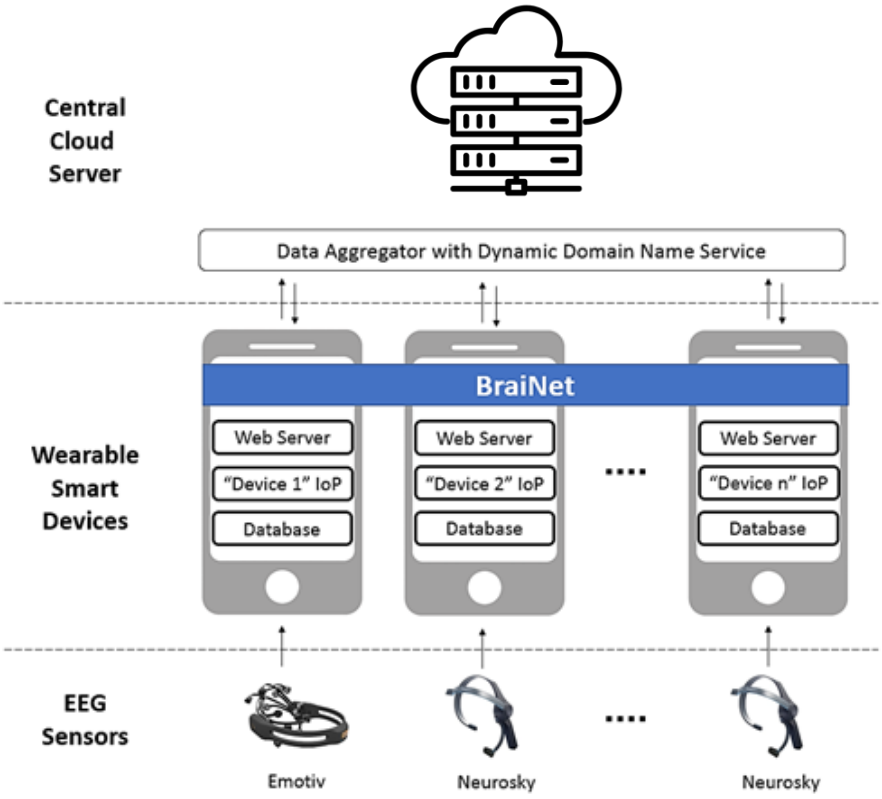
We wan to use this hierarchical communication infrastructure

- **To satisfy accuracy, real-timing, and energy efficiency of BCI apps, computational intensive signal processing is implemented in multi-tier architecture including fog and cloud servers.**



System Model

- **“BraiNet” framework consists of a middleware installed in mobile devices for developing large scale BCI apps.**

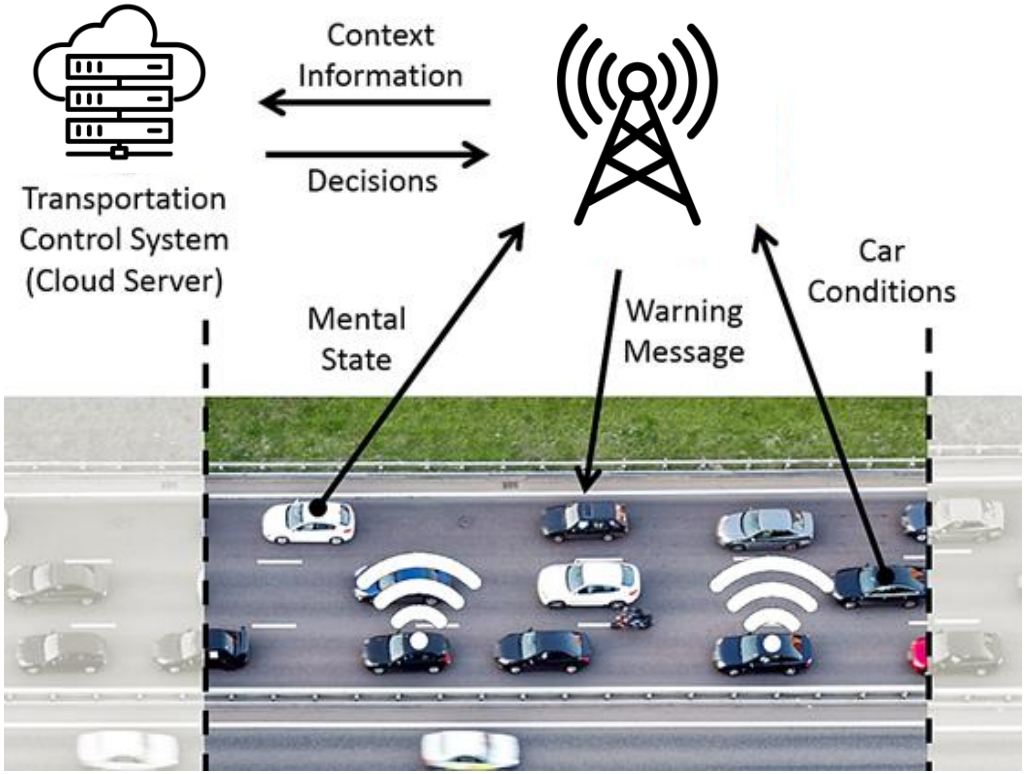


Applications

We can send driver brain sensors to cloud which can calculate attention state of drivers. So we can respond to unsafe drivers.

SafeDrive

- According to mental states of drivers, appropriate warning message will be provided to avoid accidents.

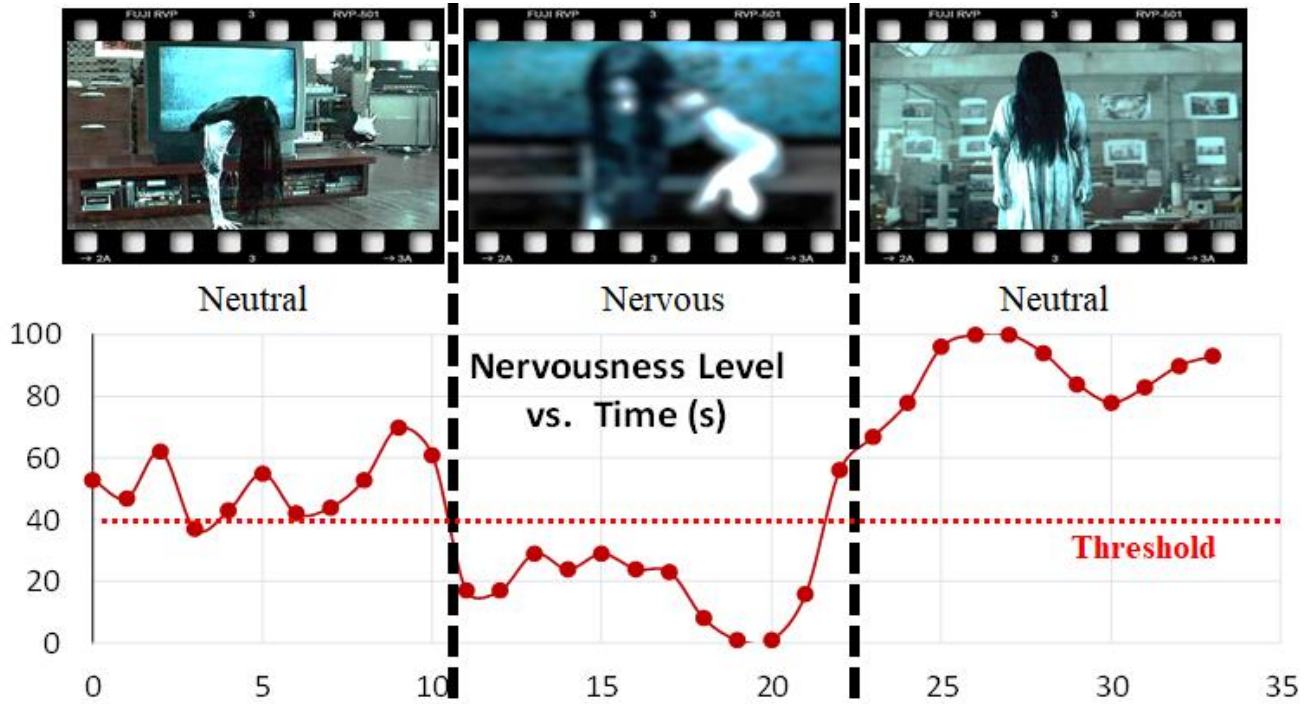


Applications

It takes data from brain and figures our nervousness level of people, then it blurs scenes with scary details

nMovie

- According to mental state of the viewer, the movie scene may change.

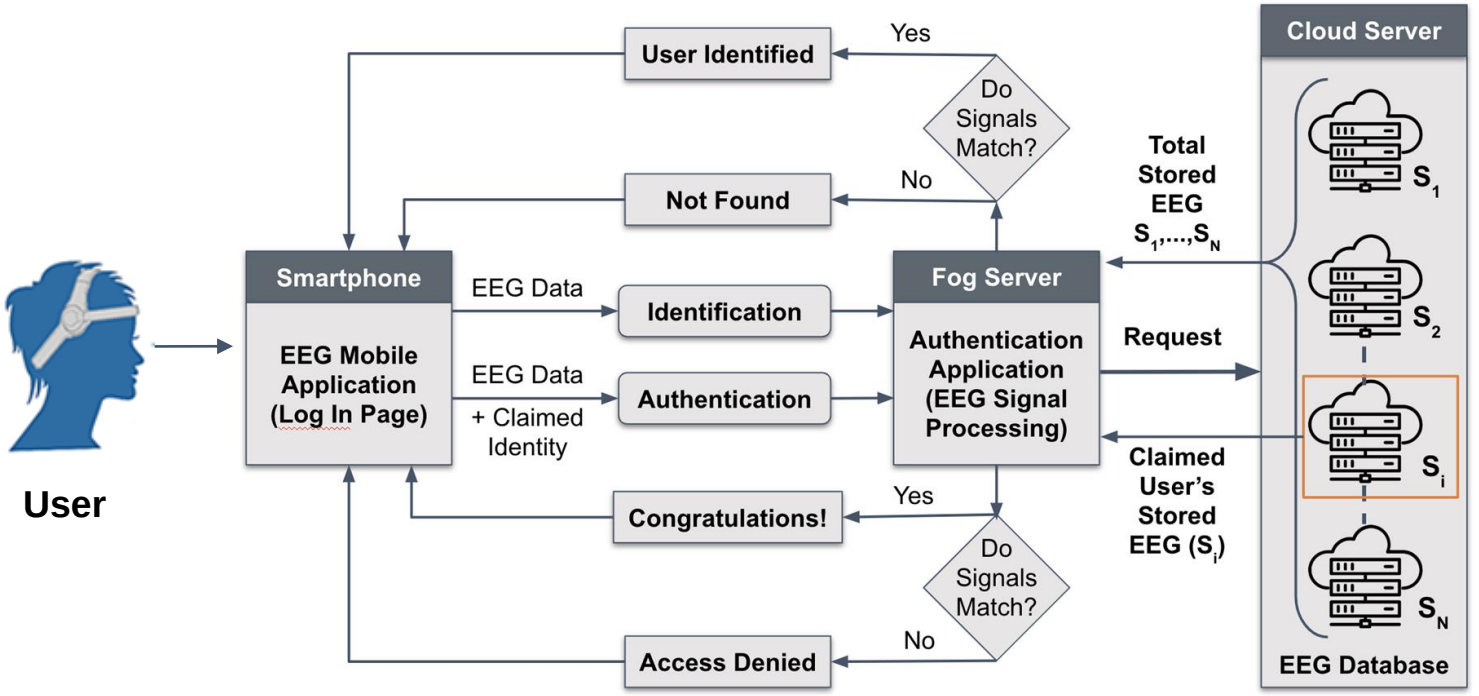


Applications

brain driven authentication - using unique thoughts become your password

E-BIAS

- Each person has unique brain signal patterns. These patterns can be considered as signatures for identification and authentication in security systems.

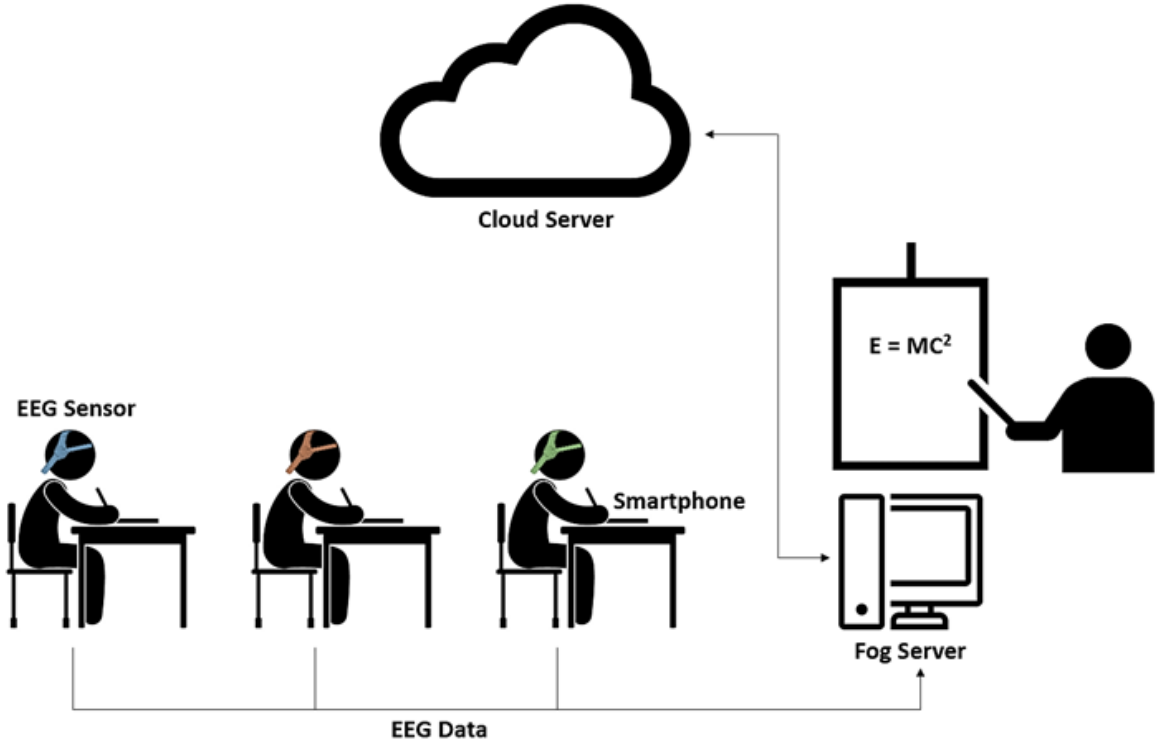


Applications

AyLA

monitoring attention level of students in classroom. If half class is dozing => may be change teaching style.

- According to students' attention level, the lecture materials may change.



Security in Mobile Applications

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Security in Mobile Banking Apps

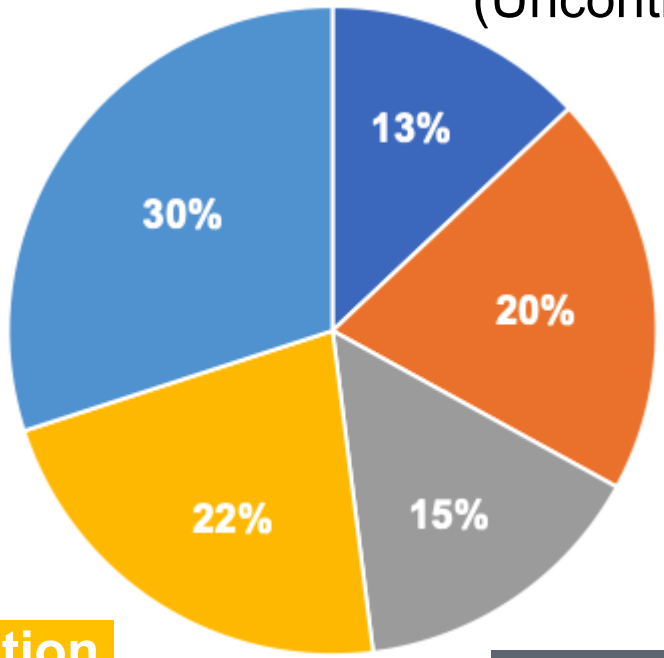
- Today's smartphones are PCs of yesteryears with Trojans, unauthorized access, and data leakage
- 90% of banking apps are found to have vulnerabilities (Cenzic report)
- On average there are 14 vulnerabilities per app
- RSA conference mobile app has security vulnerabilities
- **There is a flaw in the implementation of strong security techniques**

Security in Mobile Banking Apps

Insecure External Connectivity
(Server configuration, application level security in web services)

Conclusion is - we had solutions
But they were not implemented properly

Excessive Privileges
(Uncontrolled access)



Insufficient Input Validation
(Buffer overflow, SQL injection, firmware update, data injection)

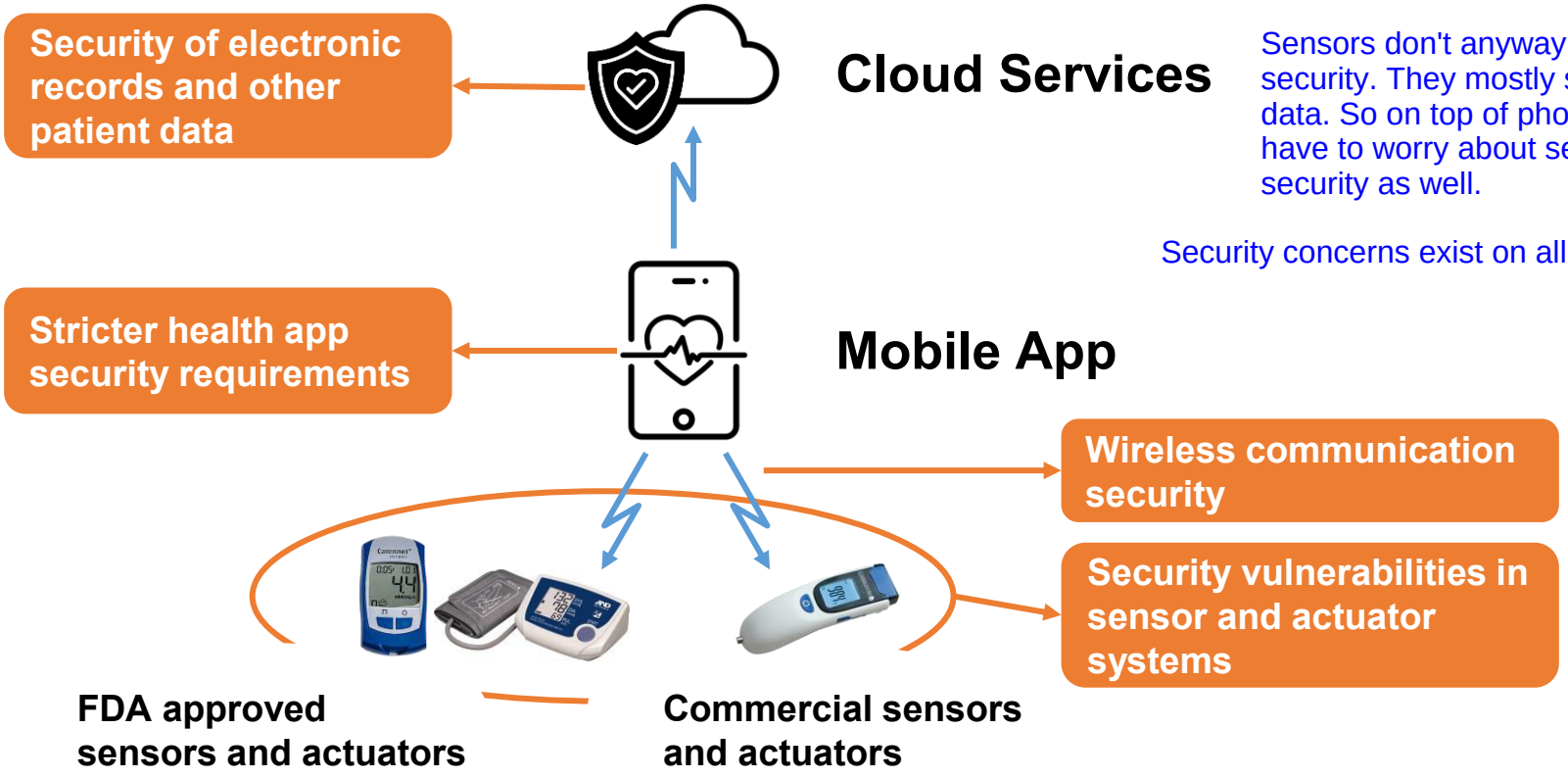
Privacy Violation
(Data leakage, lack of encryption)

Poor Session Management
(authentication to a web server, bad transport layer security)

Configuration Management Problem
Default configurations may not always be best or very secure

Security Challenges in Medical Control Apps

- Resource constraints in sensors and actuators
- Poor software development support for sensors and actuators
- Real-time requirements of Health apps



Safety Security Inter-relation

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

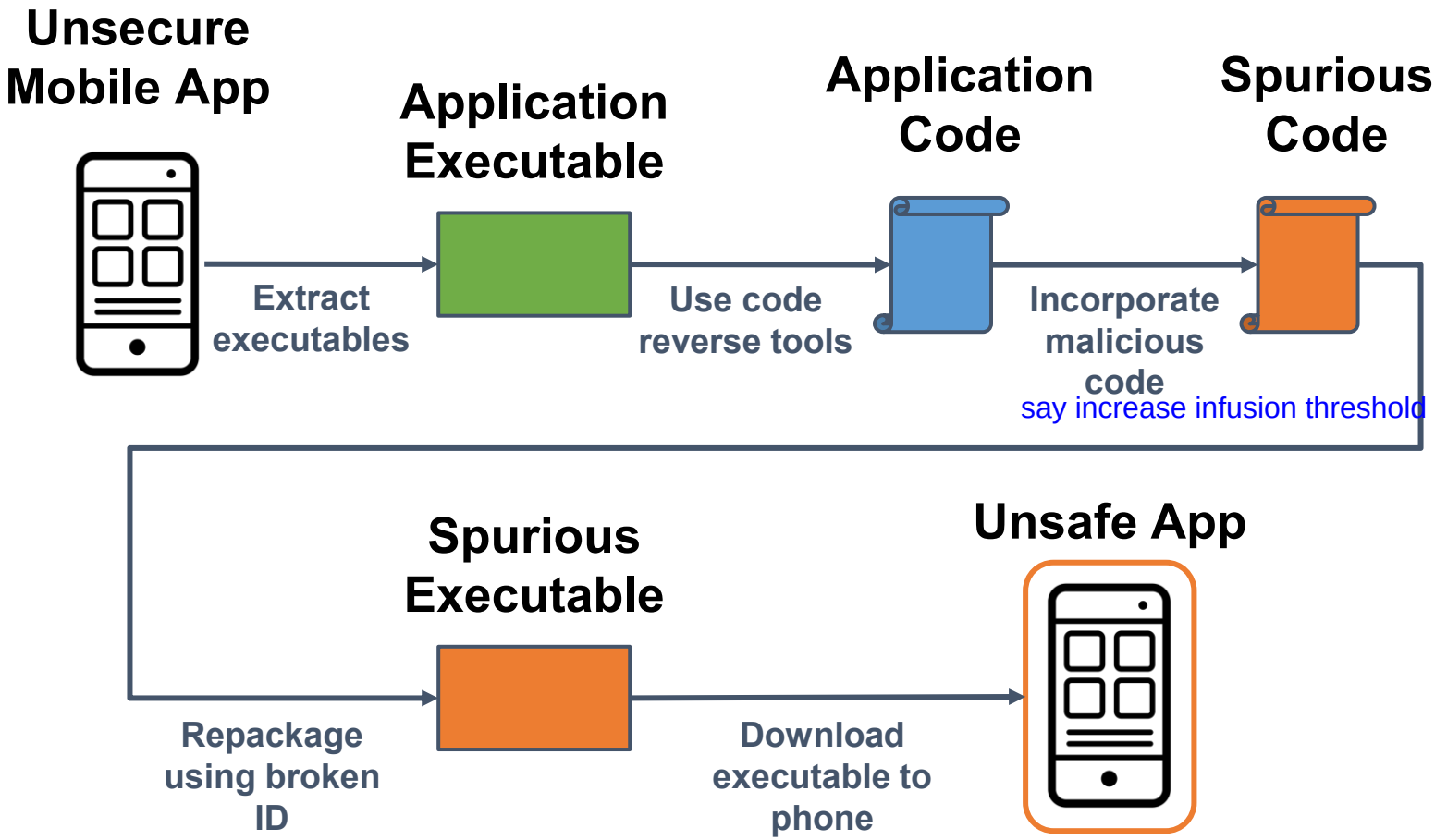
Ira A. Fulton Schools of Engineering
Arizona State University

Mobile App Safety Security Inter-Relation

Secuity have implications on human safety as well. We will how security is related to safety

Mechanism 1: Security vulnerability exploited to cause harm [Brute All]

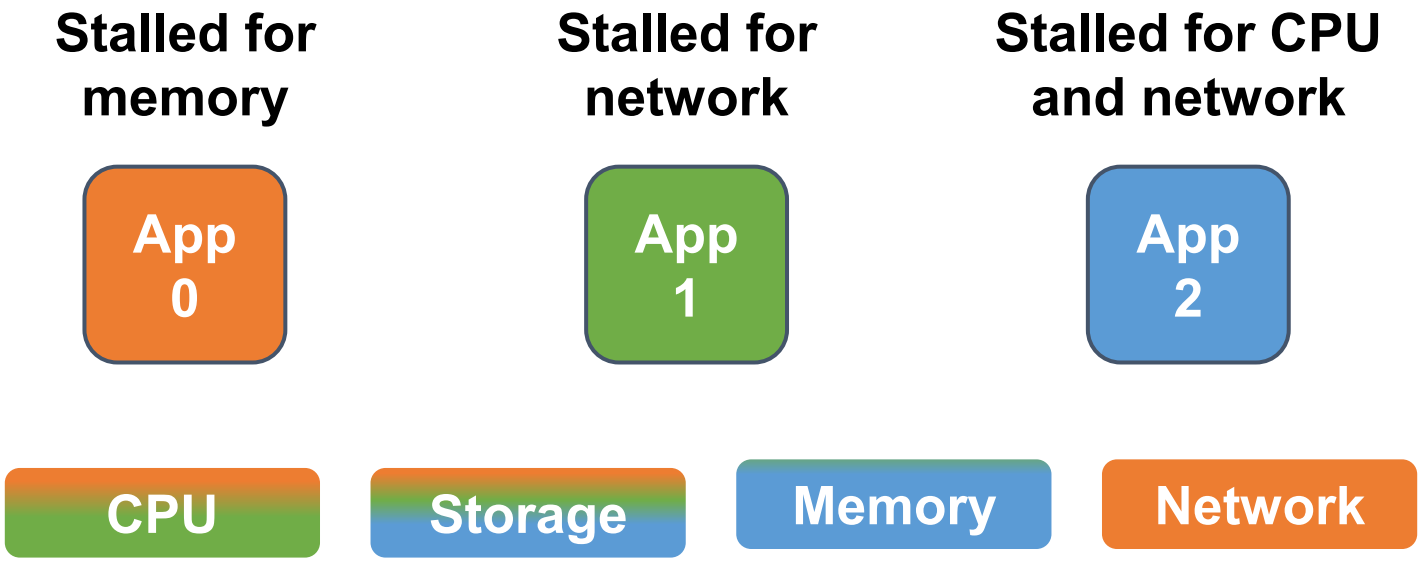
Usecase example: Artificial pancreas have insulin infusion limit to prevent hyperglycemia.



Mobile App Safety Security Inter-Relation

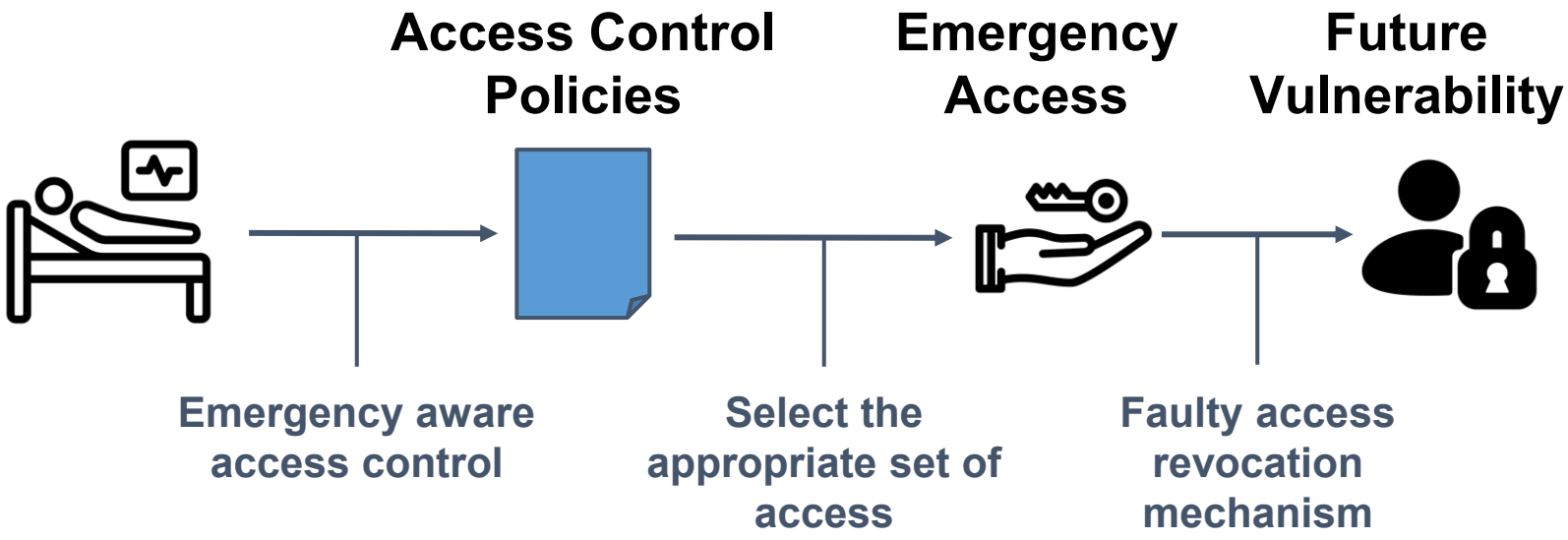
Mechanism 2: Security protocols introduce unexpected delays in control operation [Aleksandar Deep]

Example: say all commands are encrypted between phone and infusion controller. Now controller will take 10 minutes to respond, it was supposed to run within 5 minutes.



Mobile App Safety Security Inter-Relation

Mechanism 3: Improper safety assurance policies compromise security



Concept of criticality aware access control: teacher doesn't give door access to anybody. But say fire breakouts - to save lives, teacher can give everyone acces. If they forgot to revoke access, students might later misuse it.
Security compromise made for safety once can be alter exploited later by malicious entities.

Network Security Basics

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Security Basics

- **What is network security?**
- **Principles of cryptography**
- **Usage of cryptography for network security**
  - Message integrity
  - Digital signature
  - Endpoint authentication
- **Key establishment**
- **Example application**
  - Securing e-mail

What is Network Security?

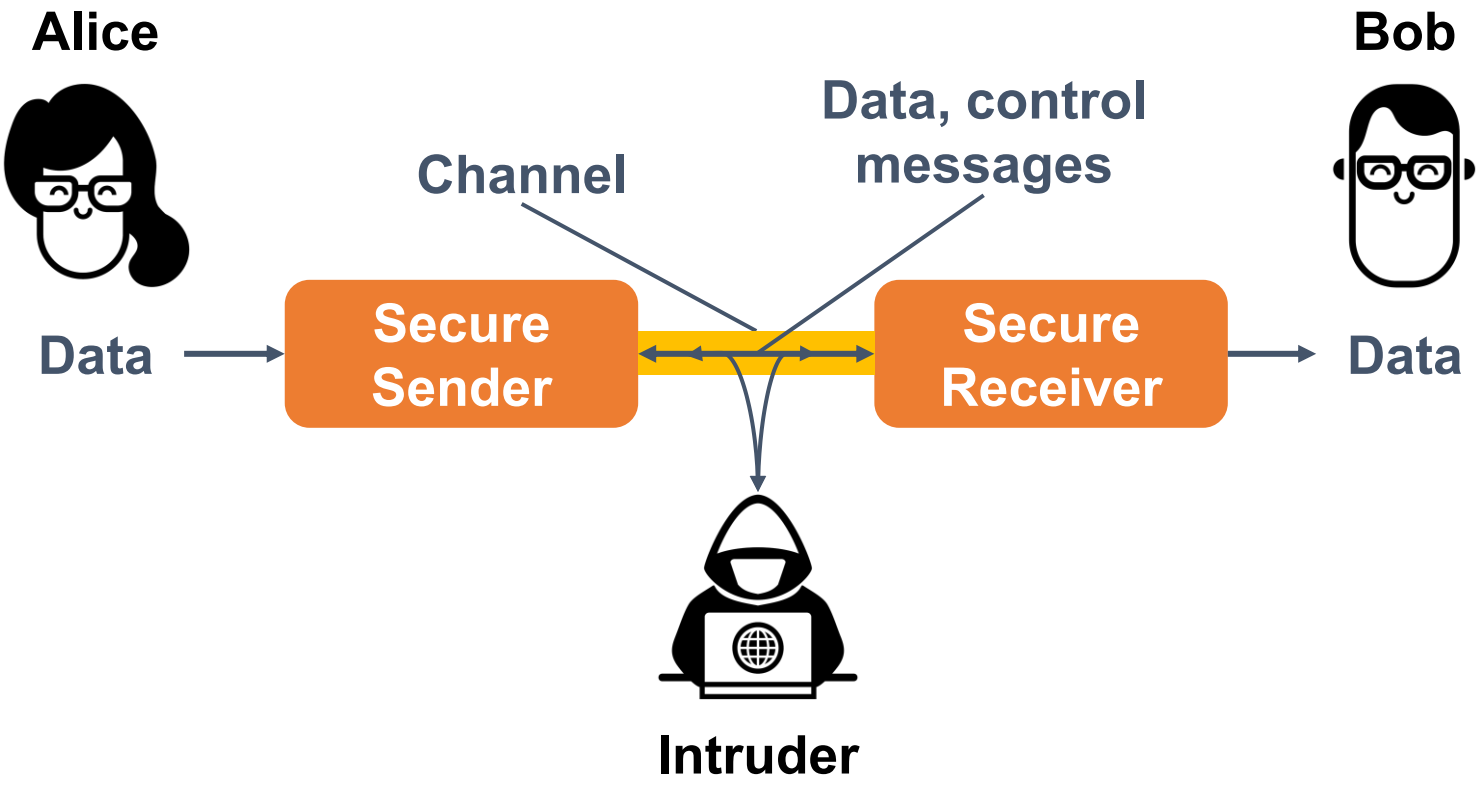
4 different pillars:

- **Confidentiality:** Only sender, intended receiver should "understand" message contents
  - Sender encrypts message
  - Receiver decrypts message
- **Authentication:** Sender, receiver want to confirm identity of each other its really intended sender itself on other end
- **Message integrity:** Sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
- **Access and availability:** Services must be accessible and available to users

No Denial of service attack

Friends and Enemies: Alice, Bob, Intruder

- Well-known in network security world
- Bob and Alice want to communicate “securely”
- Intruder may intercept, delete, add messages



Who Might Bob and Alice Be?

- **Bob and Alice can be:**
  - Real-world people
  - Web browser/server for electronic transactions (e.g. online purchases)
  - Online banking client/server
  - DNS servers
  - Routers exchanging routing table updates

Intruders

- **Question: What can an "intruder" do?**
- **Answer: A lot:**
  - Eavesdrop – intercept messages
  - Actively insert messages into the connection spurious messages
  - Impersonate – can fake (spoof) source address in packet (or any field in packet) Bob will see this intruder as if it is Alice
  - Hijack – "take over" ongoing connection by removing the sender or receiver and inserting himself or herself in that place
  - Denial of service – prevent service from being used by others (e.g. by overloading resources)

Network Security Cryptography

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Security Basics

- What is network security?
- Principles of cryptography
- Usage of cryptography for network security
  - Message integrity
  - Digital signature
  - Endpoint authentication
- Key establishment
- Example application
  - Securing e-mail

Cryptography

- **Cryptography**
  - A set of mathematical functions with a set of nice properties
  - A common mechanism for enforcing policies

Only intended sender and receiver can understand that cipher text

- **Encrypt clear text into cipher text, and vice versa**
- **Properties of good encryption techniques:**

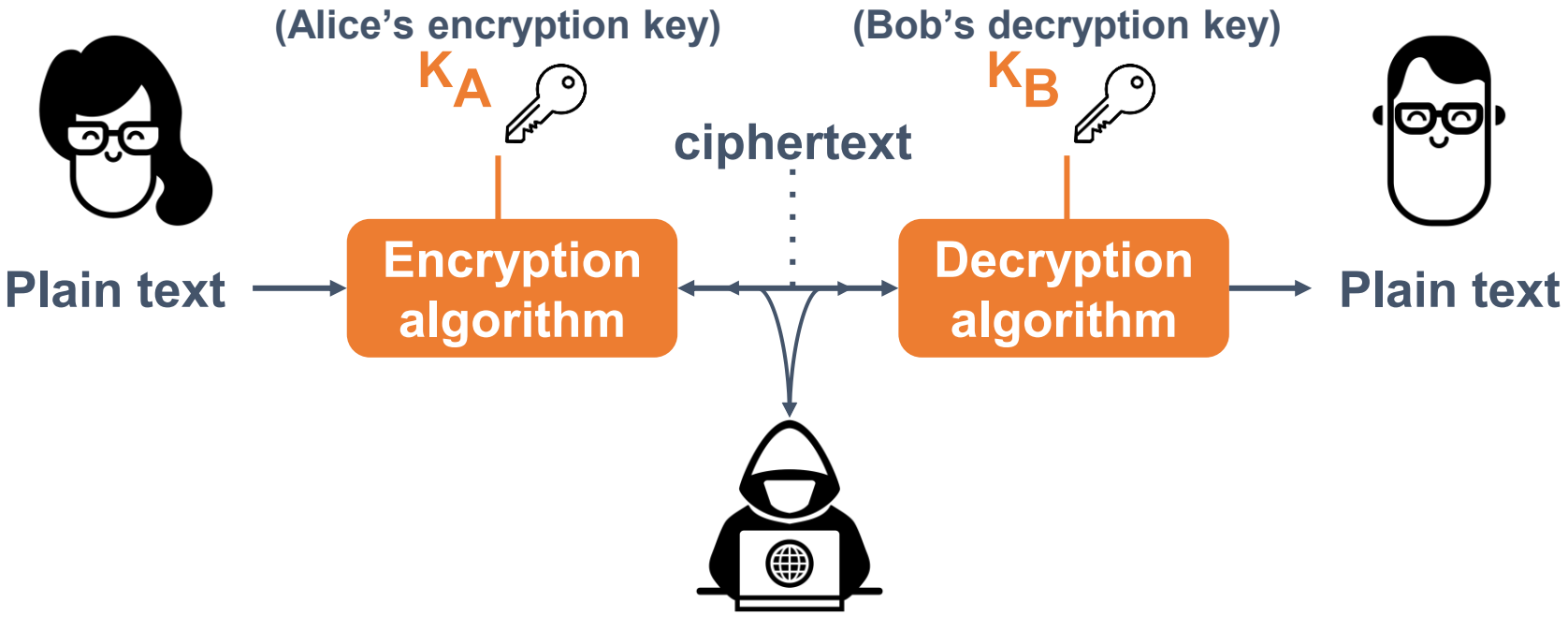
Intruder might know that you are using AES but thye should never know encryption the key

  - Encryption scheme depends not on secrecy of algorithm but on parameter of algorithm (i.e. encryption key)
  - Extremely difficult for an intruder to determine the encryption key

Cryptography Algorithms

- **Symmetric key algorithm**
  - One shared by a pair of users used for both encryption and decryption
- **Asymmetric or public/private key algorithms are based on each user having two keys:**
  - **Public key**: in public
  - **Private key**: key known only to the individual user

The Language of Cryptography



- **Symmetric key crypto:** sender and receiver keys are identical
- **Public-key crypto:** encryption key is public, decryption key is secret (private)

Symmetric Key Cryptography

Some examples:

- **Substitution cipher:** substituting one thing for another
- **Monoalphabetic cipher:** substitute one letter for another

plaintext: abcdefghijklmnopqrstuvwxyz

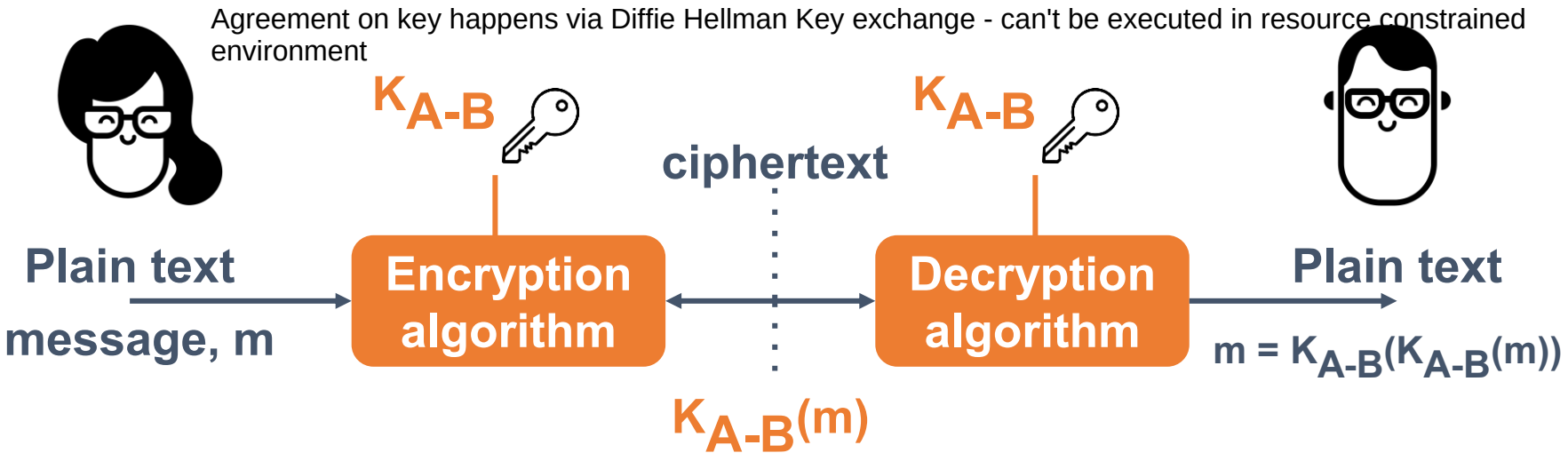
ciphertext: mnbvcxzasdfghjklpoiuytrewq

- **Example:** **plaintext:** **Bob. I see you. Alice**

  ciphertext: **nkn. s icc wky. mgsbc**

- **Question:** How hard is it to break this simple cipher?
  - Brute force (how difficult?)
  - Other? Such substitution cipher is very easy to break. Don't do brute force. There are too many combinations. Frequency analysis can help. Known-plain text attack can help.

Symmetric Key Cryptography



- **Symmetric key crypto:** Bob and Alice share the same (symmetric) key: K_{A-B}
  - E.g. key is knowing substitution pattern in monoalphabetic substitution cipher
- **Question:** How do Bob and Alice agree on key value?

Symmetric Key Crypto: DES

DES – Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input

How secure is DES?

- DES challenge: A 56-bit key-encrypted phrase (“Strong cryptography makes the world a safer place”) was decrypted (brute force) in 4 months
- No known “backdoor” decryption approach

Making DES more secure:

- Use three keys sequentially (3-DES) on each datum
- Use cipher-block chaining

AES: Advanced Encryption Standard

- New (November 2001) symmetric-key NIST standard replacing DES
- Processes data in 128-bit blocks
- 128-, 192-, or 256-bit keys
- Brute force decryption (trying each key) takes 1 second on DES and 149 trillion years for AES

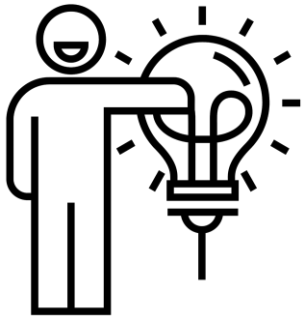
big jump

Public Key Cryptography

Symmetric key crypto

- Requires the sender and receiver to know the shared secret key

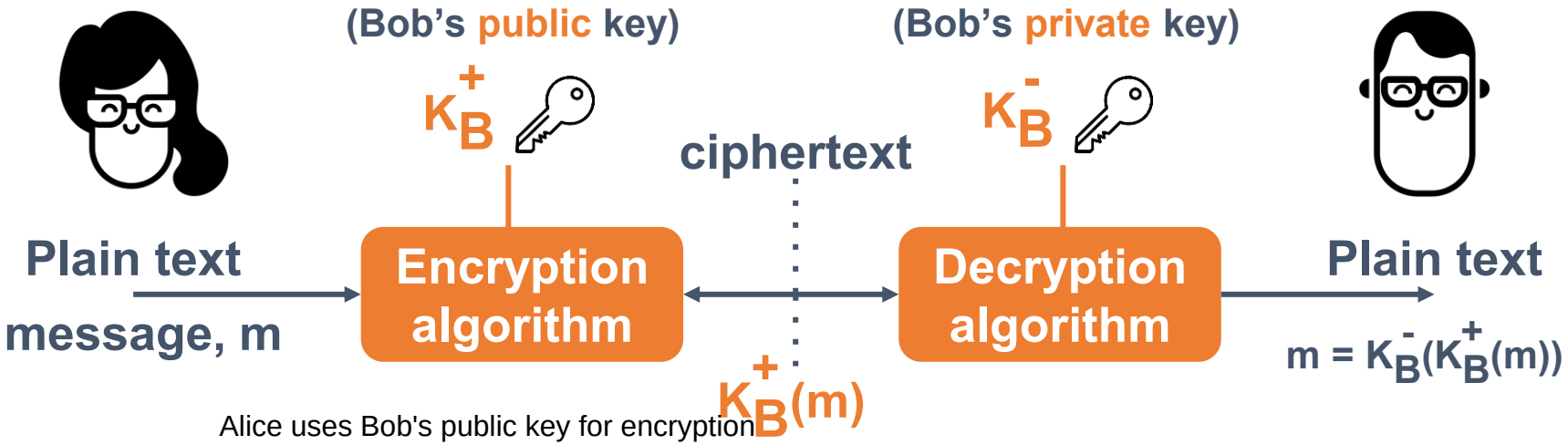
Question: How do you agree on a key in the first place (particularly if never "met")?



Public key cryptography

- Radically different approach [Diffie-Hellman76, RSA78]
- The sender and receiver do **not** share a secret key
- The **public** encryption key is known to **all**
- The **private** decryption key is known only to the receiver

Public Key Cryptography



Alice uses Bob's public key for encryption

Public Key Encryption Algorithms

- **Requirements:**
  - Need $K_B^+(\bullet)$ and $K_B^-(\bullet)$ such that:

$$K_B^-(K_B^+(m)) = m$$

  - Given public key K_B^+, it should be impossible to compute private key K_B^-
- **RSA: Rivest, Shamir, Adleman algorithm**

RSA is public key algorithm

RSA: Choosing Keys

We want to highlight how difficult RSA is and why it can't be implemented in a sensor.

- **Choose two large prime numbers *p, q***
  - E.g. 1024 bits each

To check if p is prime, we need to divide it by all numbers from 0 to root p and remainder must be zero.

- **Compute *n* = pq, z = (p-1)(q-1)**

if p is 1024 bits, it can take 2^1024 values and thus checking if it is prime needs 2^512 complexity
=> first step itself is exponential. Impossible for sensor

- **Choose *e* (with e < n) that has no common factors with z**
  - *e, z* are “relatively prime”
- **Choose *d* such that *ed-1* is exactly divisible by z**
  - In other words: *ed* mod z = 1
- **Public key is (*n,e*) and Private key is (*n,d*)**

K_B^+ K_B^-

RSA: Encryption, Decryption

- **Given (*n*,*e*) and (*n*,*d*) as computed above**
- **To encrypt bit pattern, *m*, compute:**

  $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)
- **To decrypt received bit pattern, c, compute:**

  $c = m^e \bmod n$ (i.e., remainder when c^d is divided by n)

$$m = \underbrace{(m^e \bmod n)^d}_{c} \bmod n$$

RSA Example

- **Bob chooses:** Here we chose only 3 bit long primes, forget 1024 bits
  - p = 5, q = 7; then n = 35, z = 24
  - e = 5 (so *e, z* relatively prime)
  - d = 29 (so *ed-1* exactly divisible by z)

Encrypt	**letter**	**m**	**m^e**	**$c = m^e \bmod n$**
	l	12	1524832	17
Decrypt	**c**	**c^d**	**$m = c^d \bmod n$**	**letter**
	17	481968572106 750915091411 825223071697	12	l

our sensors can't even store this large number
forget encryption decryption

RSA: Another Important Property

- The following property will be **very** useful later:

their order doesn't matter

$$\underbrace{K_B^-(K_B^+(m))}_{\text{Use public key first, followed by private key.}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{Use private key first, followed by public key.}}$$

The result is the same

Network Security
Message Integrity

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

ASU Ira A. Fulton Schools of Engineering
Arizona State University

Security Basics

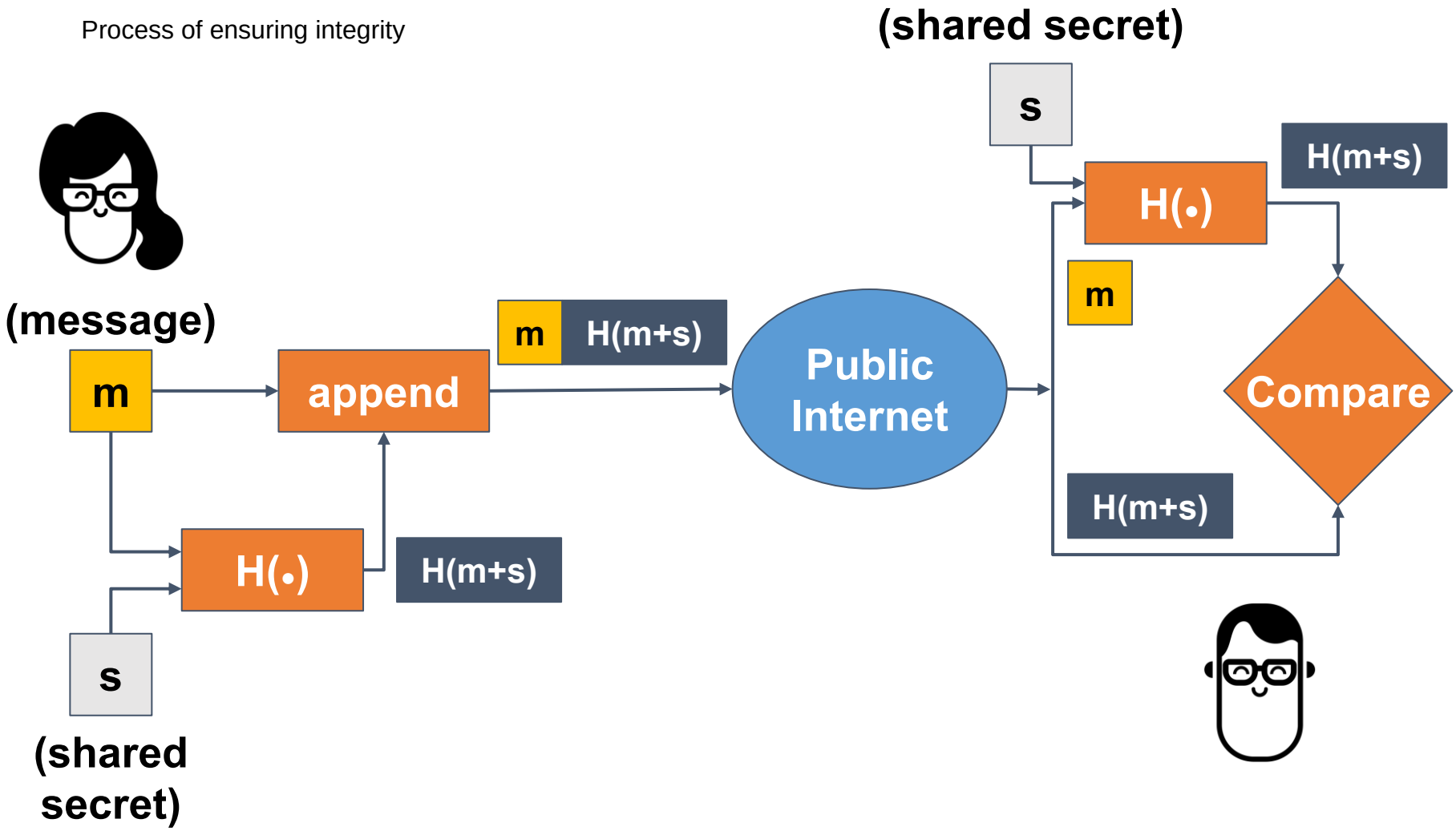
- What is network security?
- Principles of cryptography
- Usage of cryptography for network security
  - Message integrity
  - Digital signature
  - Endpoint authentication
- Key establishment
- Example application
  - Securing e-mail

Message Integrity

- **Bob receives a message from Alice and wants to ensure:**
  - The message originally came from Alice
  - The message has not changed since it was sent by Alice
- **Cryptographic hash:** we will use this to achieve integrity
  - Takes input m and produces fixed length value **H(m)**
  - Computationally infeasible to find two different messages, x, y, such that **H(x) = H(y)**
    - Equivalently: Given m = H(x), (x unknown), you can not determine x

in general, if x!=y then H(x)!=H(y)
But this might not always hold true because of possible collisions.

Message Authentication Code



MACs in Practice

- **MD5 hash function widely used [RFC 1321]**
  - Designed by Rivest, 1992
  - Computes 128-bit MAC in a 4-step process
  - Arbitrary 128-bit string **x** – appears difficult to construct message **m** whose MD5 hash is equal to **x**
    - Recent (2005) attacks on MD5
- **SHA-1 is also used**
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit MAC
  - Possible to break it faster than brute force – 2005

Network Security
Digital Signatures

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Security Basics

- What is network security?
- Principles of cryptography
- Usage of cryptography for network security
  - Message integrity
  - Digital signature
  - Endpoint authentication
- Key establishment
- Example application
  - Securing e-mail

Digital Signatures

Used for establishing identity

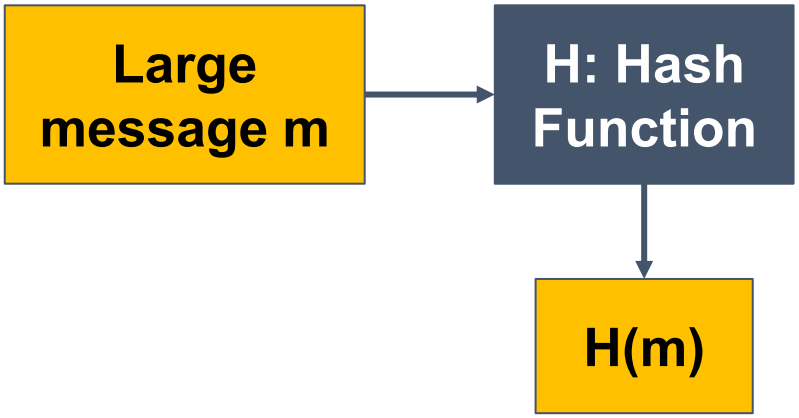
- **Cryptographic technique analogous to hand-written signatures**
  - Sender (Bob) digitally signs the document, establishing he is the document owner/creator
  - **Verifiable, non-forgeable:** Recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed the document

Message Digests

How to have digital signature?

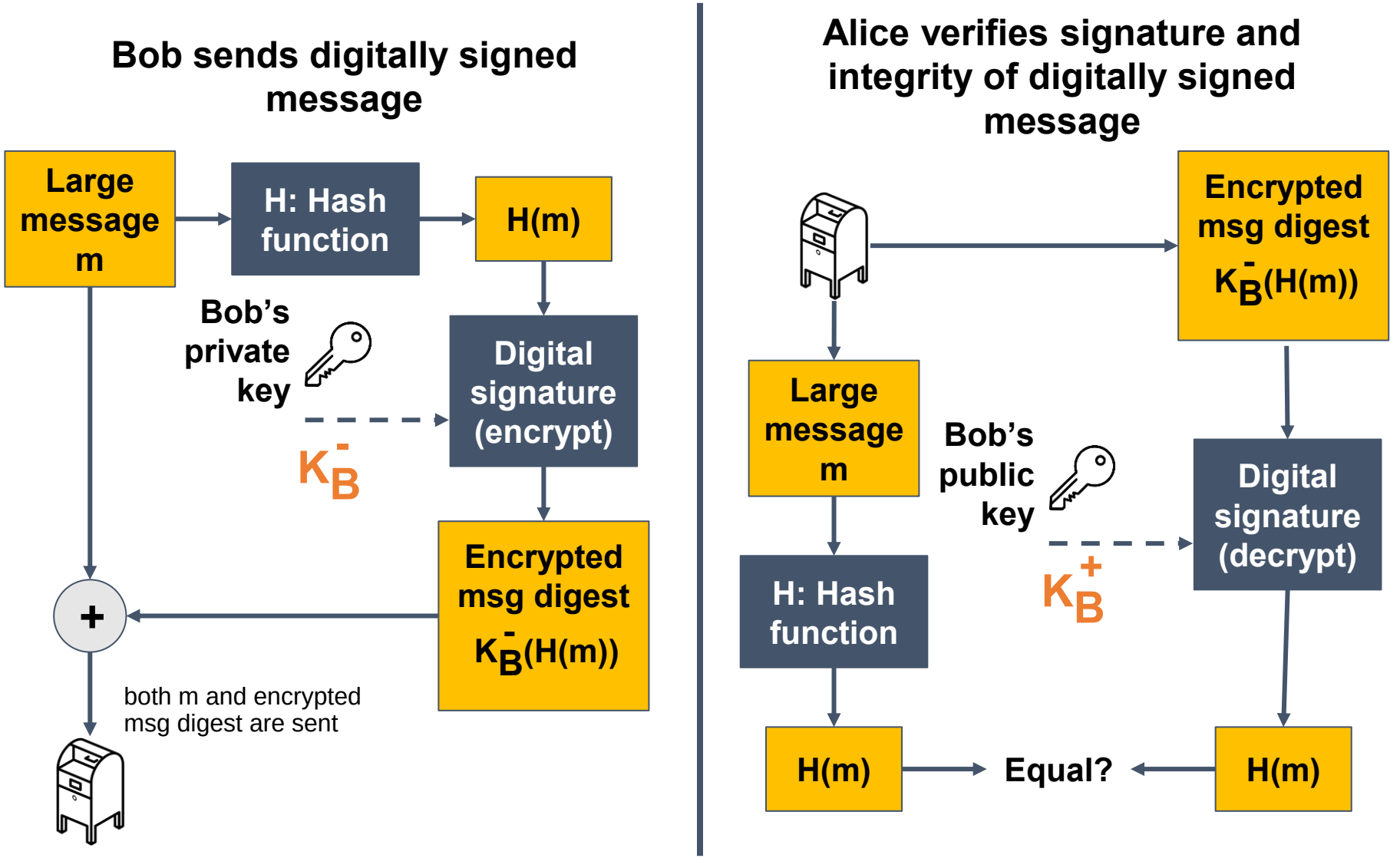
- **Computationally expensive to public-key-encrypt long messages**
- **Goal:**
  - Fixed-length, easy-to-compute digital "fingerprint"
    - Apply hash function **H** to **m** and get a fixed size message digest – **H(m)**

Applying encryption to long document is difficult
Make it a digest, apply hash function to reduce message size. Some collision is acceptable.



- **Hash function properties:**
  - Many-to-1
  - Produces fixed-size message digest (fingerprint)
  - Given message digest **x**, computationally infeasible to find **m** such that **x = H(m)**

Digital Signature = Signed Message Digest



Network Security
Authentication

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

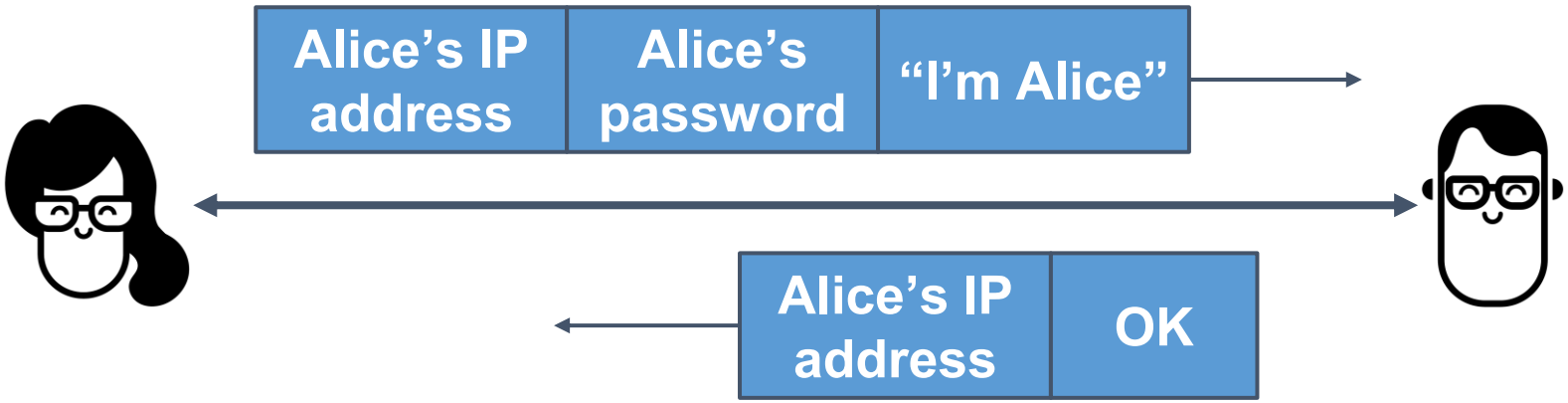
Ira A. Fulton Schools of Engineering
Arizona State University

Security Basics

- What is network security?
- Principles of cryptography
- Usage of cryptography for network security
  - Message integrity
  - Digital signature
  - Endpoint authentication
- Key establishment
- Example application
  - Securing e-mail

Authentication: A Naïve Approach

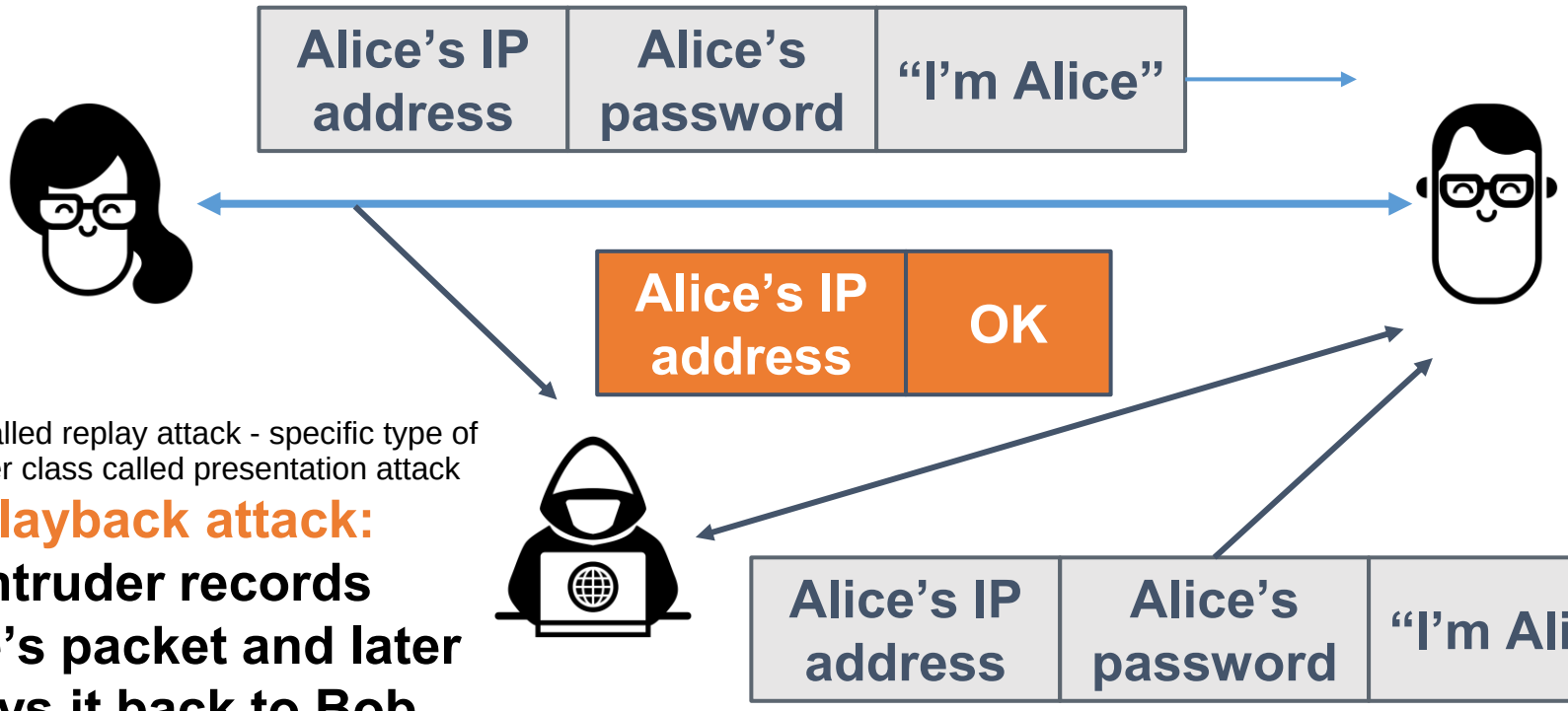
- **Goal:** Guarantee that a party is who it claims he/she is.
- **Protocol:** Alice says "I am Alice" and sends her secret password to "prove" it.



Failure scenario?

Authentication: A Naïve Approach

Protocol: Alice says "I am Alice" and sends her secret password to "prove" it.



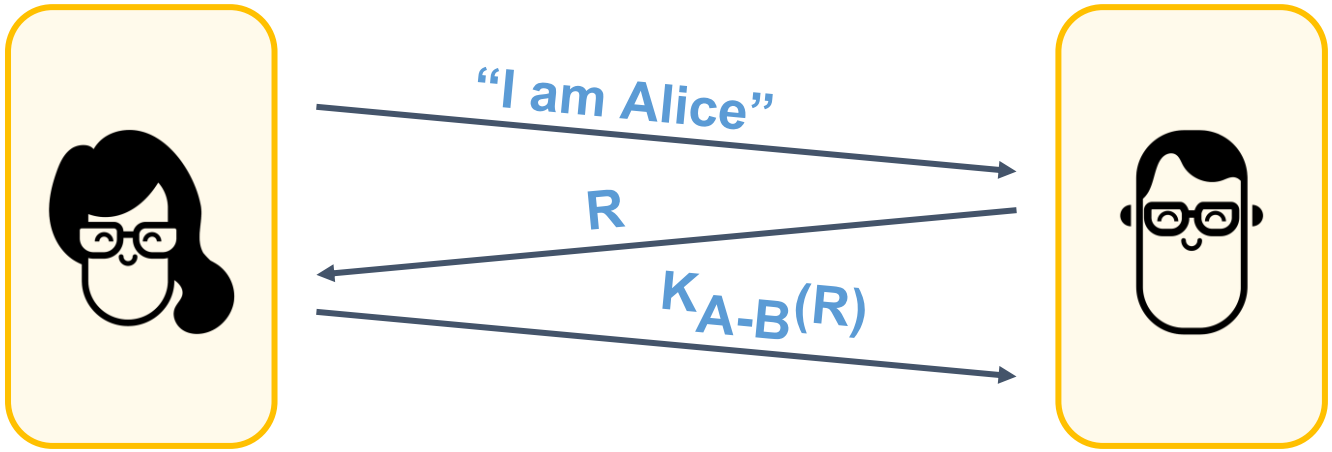
also called replay attack - specific type of broader class called presentation attack

Playback attack:

Intruder records Alice's packet and later plays it back to Bob

Authentication: Symmetric Key

- **Goal:** Avoid playback attack (we are using symmetric key here)
- **Nonce:** Number (R) used only once-in-a-lifetime
  random number
- **Protocol:** To prove Alice is "live", Bob sends Alice a nonce (R). Alice must return R, encrypted with the shared secret key

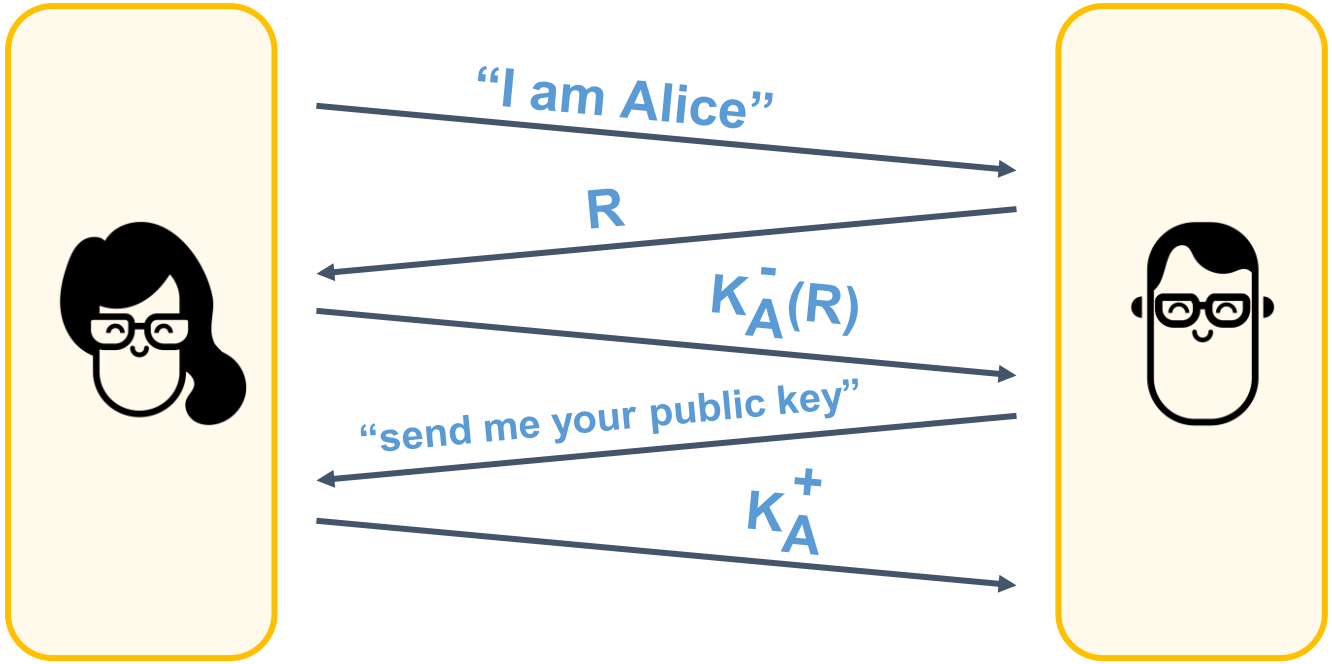


Alice is live, and only Alice knows the key to encrypt nonce, so it must be Alice

Failures, drawbacks?

Even if intruder captures encrypted random number sent by Alice and replay it later, Bob knows that this number is repeated - which is not allowed

Authentication: Public/Private Key



Bob computes

$K_A^+(K_A^-(R)) = R$

and knows only Alice could have the private key, that encrypted R such that

$K_A^+(K_A^-(R)) = R$

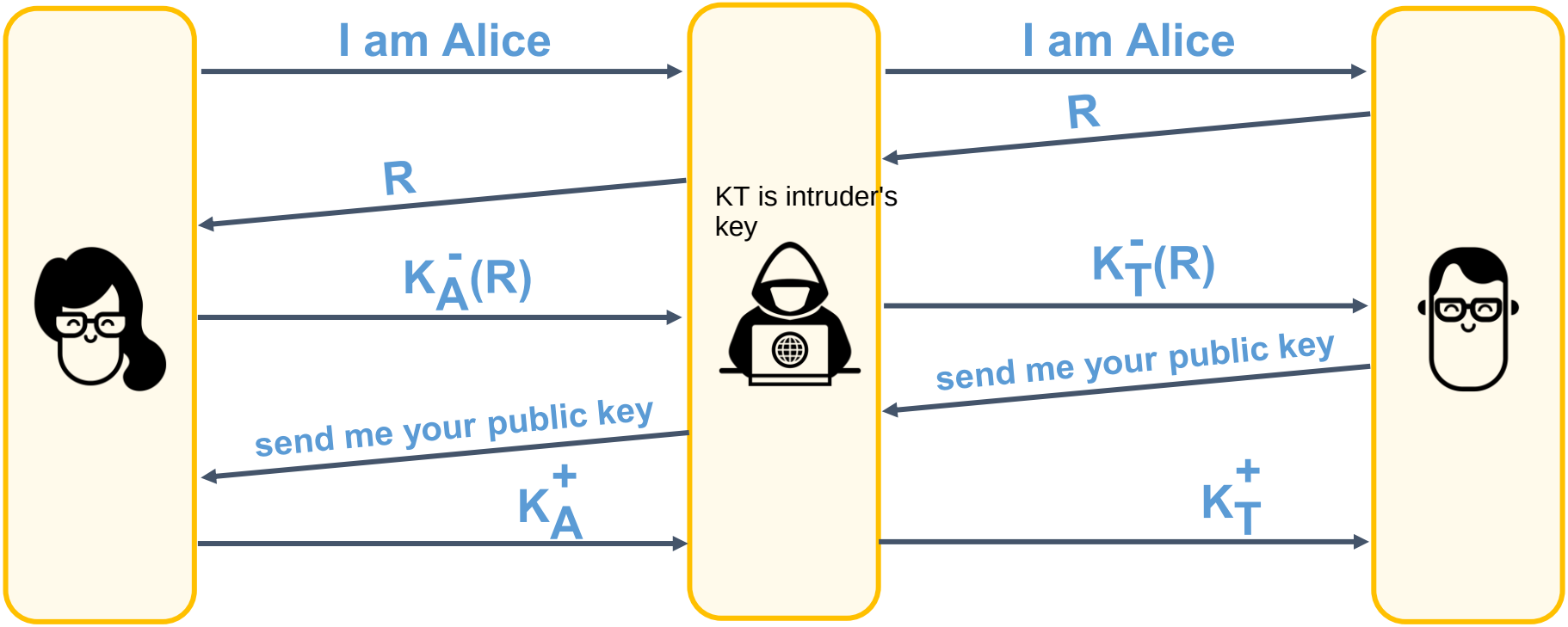
KA is Alice's key; KA- is ALice's private key, + is public

Failures, drawbacks?

Man in the Middle

special type of impersonation attack

Alice and Bob both think they are talking with each other but they are only talking with intruder



Intruder gets

$m = K_T^-(K_T^+(m))$

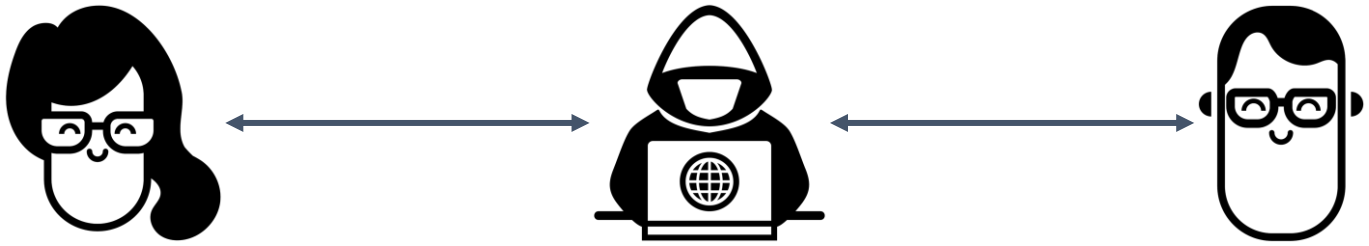
and sends m to Alice encrypted with Alice's public key

$K_A^+(m)$

$m = K_A^-(K_A^+(m))$

$K_T^+(m)$

Man in the Middle



- **Difficult to detect:**
  - Bob receives everything that Alice sends, and vice versa. (e.g. Bob and Alice can meet one week later and recall the conversation)
  - The problem is that the intruder receives all of the messages as well

Network Security Key Exchange

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Security Basics

- What is network security?
- Principles of cryptography
- Usage of cryptography for network security
  - Message integrity
  - Digital signature
  - Endpoint authentication
- Key establishment
- Example application
  - Securing e-mail

Key Establishment

We will see how this key excahnge requires external parties and is simply not feasible in Mobile computing scenario

Symmetric key problem:

- How do two entities establish a shared secret key in the first place?

Solutions:

- Deffie-Hellman
- Trusted key distribution center (KDC) acting as intermediary between entities

Public key problem:

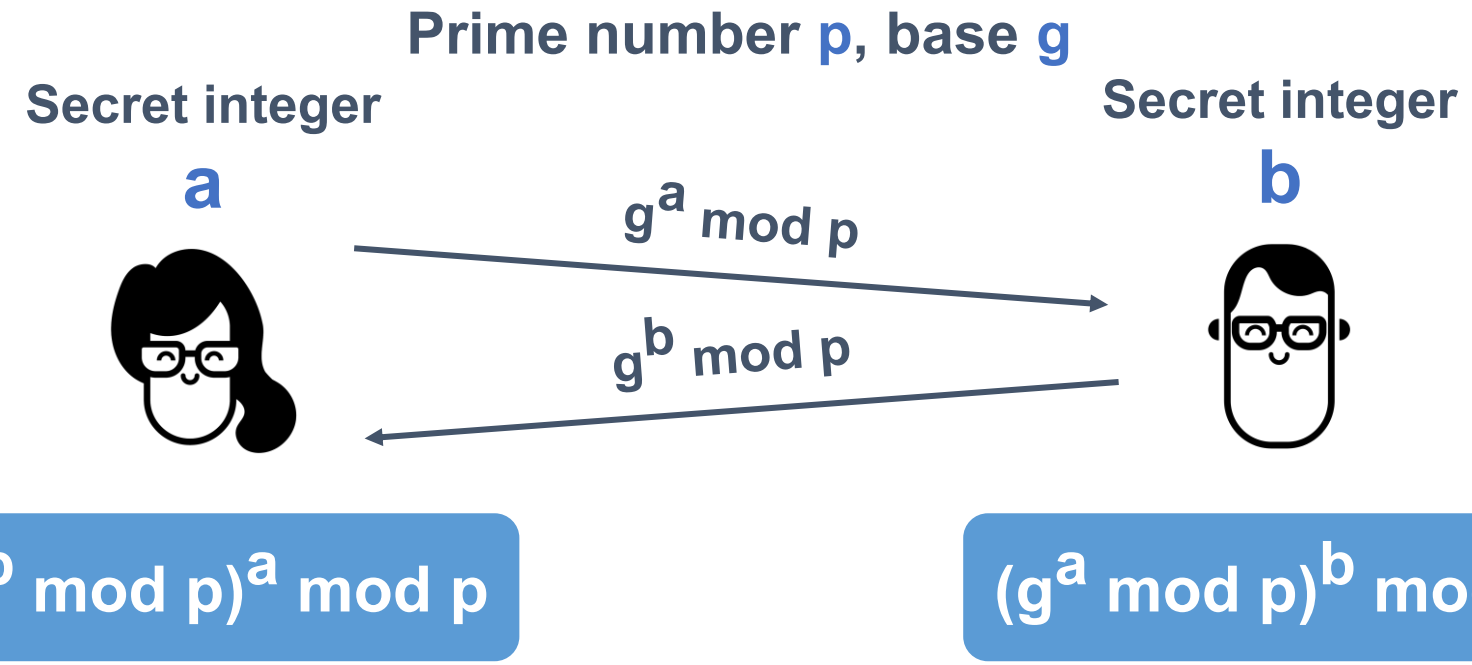
- When Bob obtains Alice's public key (from a website, e-mail, diskette), how does he know it is Alice's public key and not an intruder's?

Solution:

- Trusted certification authority (CA)

Deffie-Hellman Key Exchange

Used for symmetric key



Because of high computation required, it can't be used in sensor or even phones.

Key: $(g^b \bmod p)^a \bmod p = (g^a \bmod p)^b \bmod p$

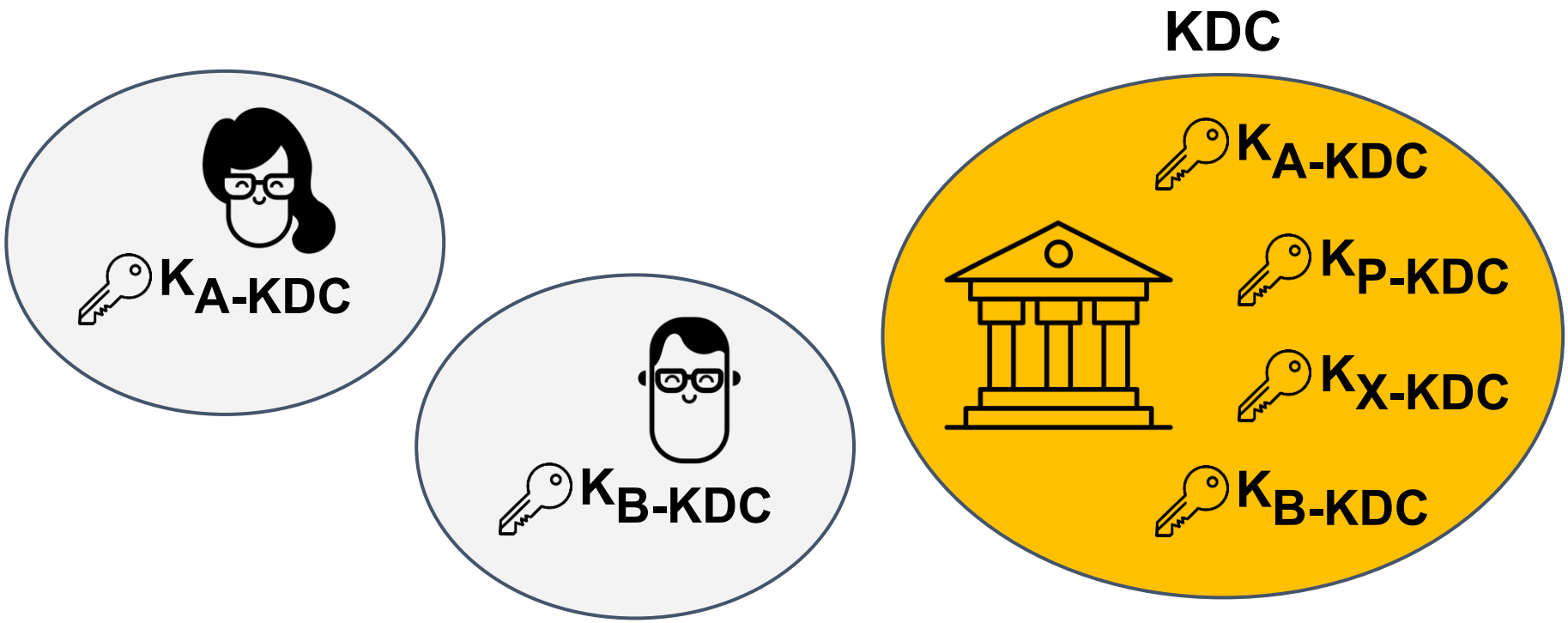
Deffie-Hellman Key Exchange: Example

- **Prime number p = 23, base g = 5**
- **Alice: a = 6**
  - Send Bob: $g^a \bmod 23 = 8$
- **Bob: b = 15**
  - Send Alice: $g^b \bmod 23 = 19$
- **Alice compute: $19^6 \bmod 23 = 2$**
- **Bob compute: $8^{15} \bmod 23 = 2$**

These numbers are very small and thus very easy to break
In practice we would not very large numbers

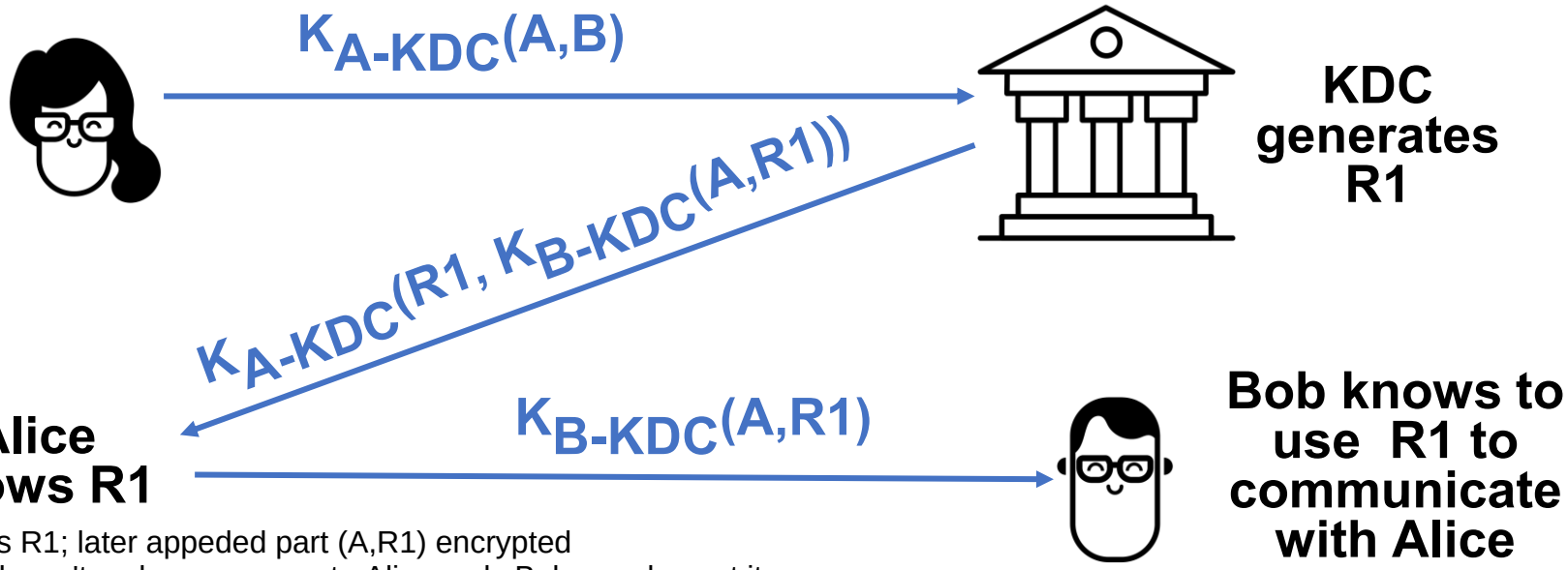
Key Distribution Center (KDC)

- **KDC:** Server shares a different secret key with each registered user (many users)
- Alice shares a key with KDC: $K_{A\text{-}KDC}$
- Bob shares a key with KDC: $K_{B\text{-}KDC}$



KDC

Question: How does a KDC allow Bob and Alice to determine a shared symmetric secret key to communicate with each other?



Deffie-Hellman vs. KDC

- **Deffie-Hellman**
  - **Pro:** No infrastructure support
  - **Con:** Computation load on users
- **KDC**
  - **Con:** Needs infrastructure support
  - **Con:** Single bottleneck, single point of failure
  - **Pro:** Computation load centered at KDC
- **Question: Are these two approaches suitable for sensor networks?**

Even such infrastructures are not feasible for MC sensors because they work in very resource constrained environment.

Network Security Certification Authority

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

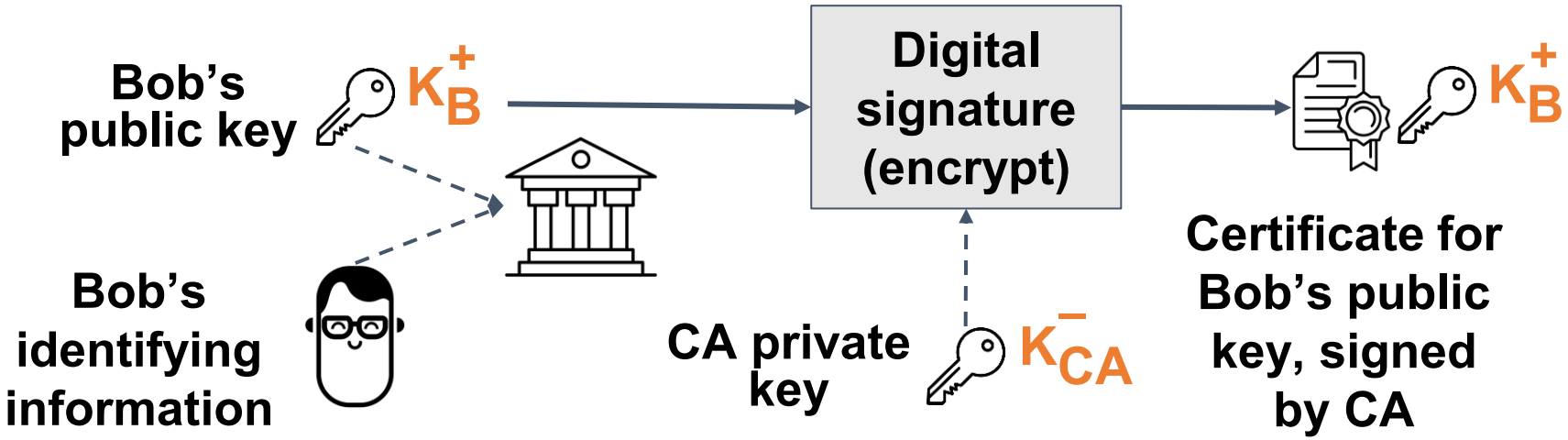
Ira A. Fulton Schools of Engineering
Arizona State University

Certification Authorities

Will see how public-private key agreement happens?

- **Certification Authority (CA): Binds public key to particular entity, E**
- **E registers its public key with the CA**
  - E provides “proof of identity” to CA
  - CA creates certificate binding E to its public key
  - Certificate containing E’s public key is digitally signed by the CA – the CA says, “This is E’s public key.”

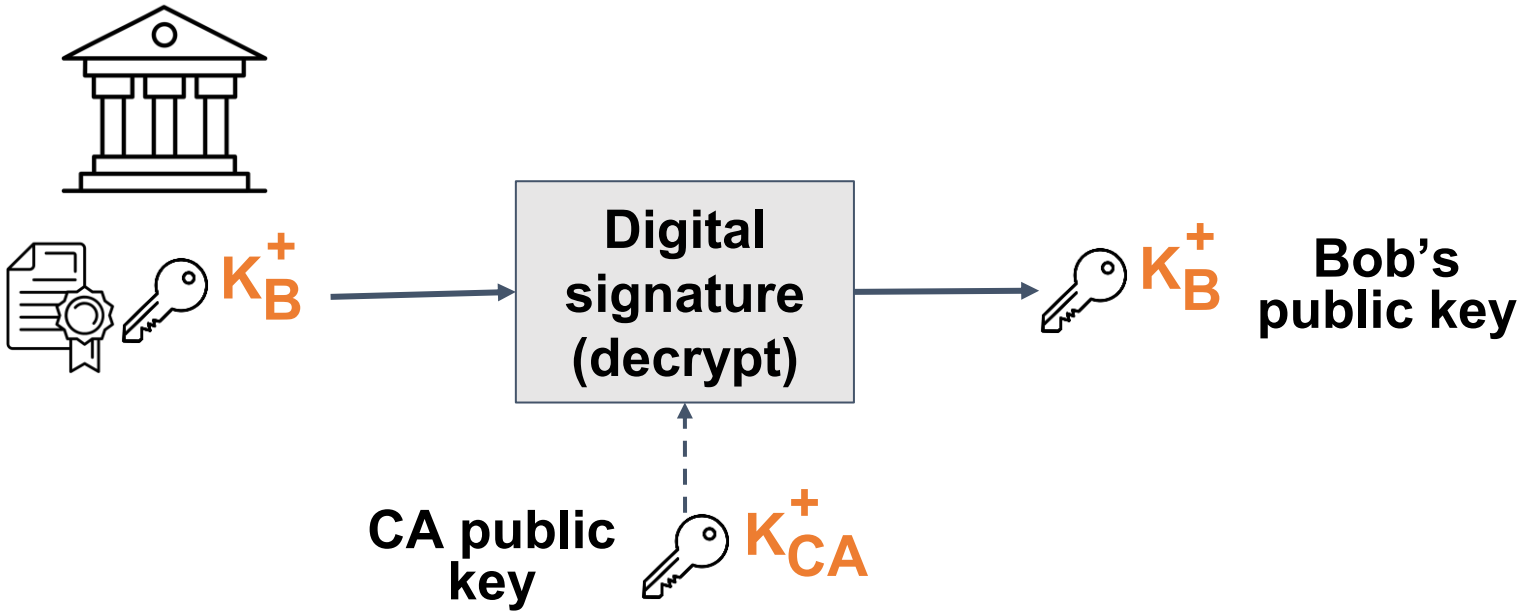
CA create public private key pair for Bob
- Gives private key to Bob to keep it secret
- Creates a digital signature for Bob's public key singed with their own private key.



Everyone has CA's public key

Certification Authorities

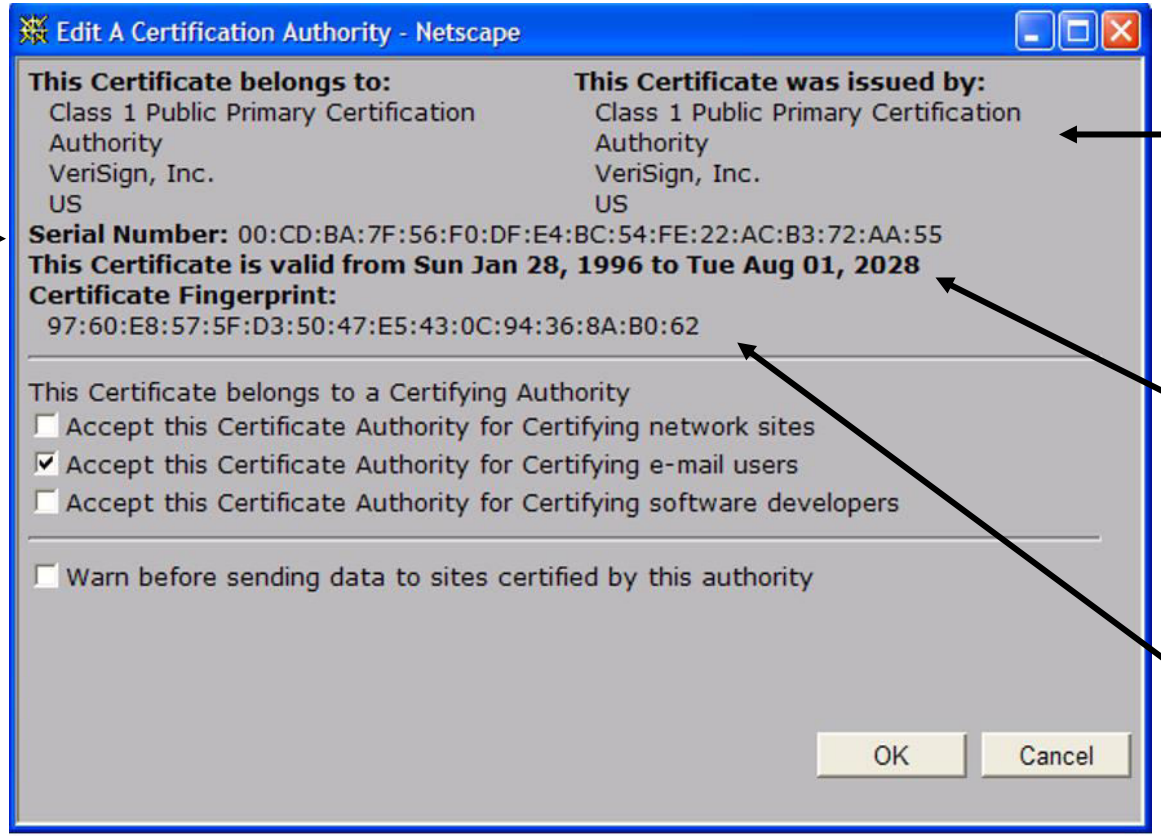
- **When Alice wants Bob's public key:**
  - Gets Bob's certificate (Bob or elsewhere)
  - Applies CA's public key to Bob's certificate to get Bob's public key



A certificate contains:

Info about certificate owner, including algorithm and key value itself (not shown)

Serial number (unique to issuer)



Info about certificate issuer

Valid dates

Digital signature by issuer

Network Security Email Security Example

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Security Basics

- What is network security?
- Principles of cryptography
- Usage of cryptography for network security
  - Message integrity
  - Digital signature
  - Endpoint authentication
- Key establishment
- Example application
  - Securing e-mail

Secure E-mail

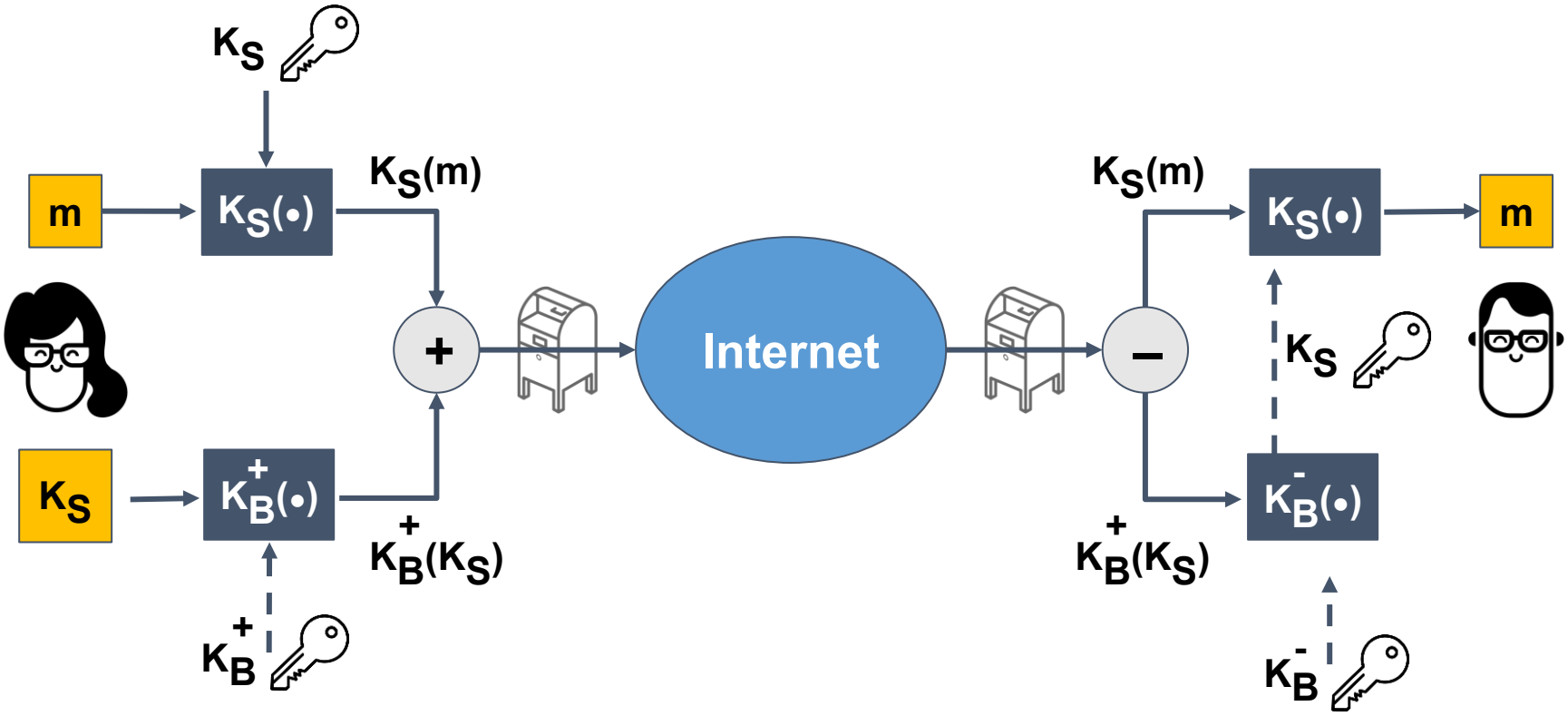
- **Alice wants to send an e-mail to Bob**
- **Requirements:**
  - Message confidential
  - Message integrity
  - Sender authentication

Confidentiality, Sender

- **Alice wants to send confidential e-mail, m, to Bob**
- **Alice:**
  - Generates random symmetric private key, KS
  - Encrypts message with KS (for efficiency)
  - Also encrypts KS with Bob's public key
  - Sends both KS(m) and KB(KS) to Bob
- **Bob:**
  - Uses his private key to decrypt and recover KS
  - Uses KS to decrypt KS(m) to recover m

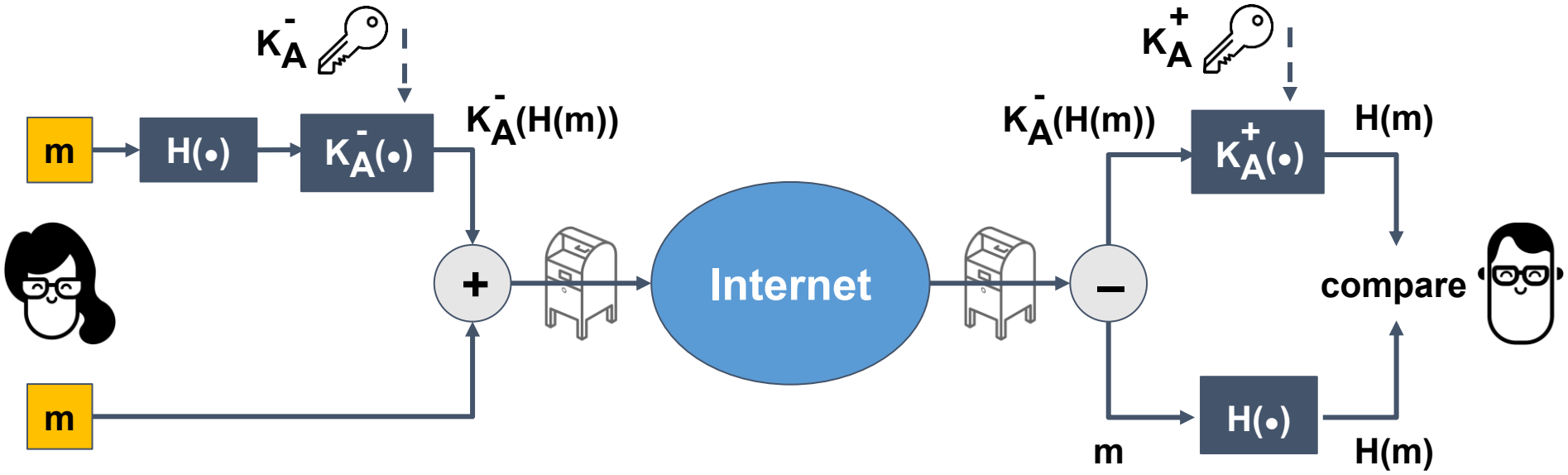
Confidentiality, Sender

- Alice wants to send confidential e-mail, m, to Bob



Sender Authentication and Message Integrity

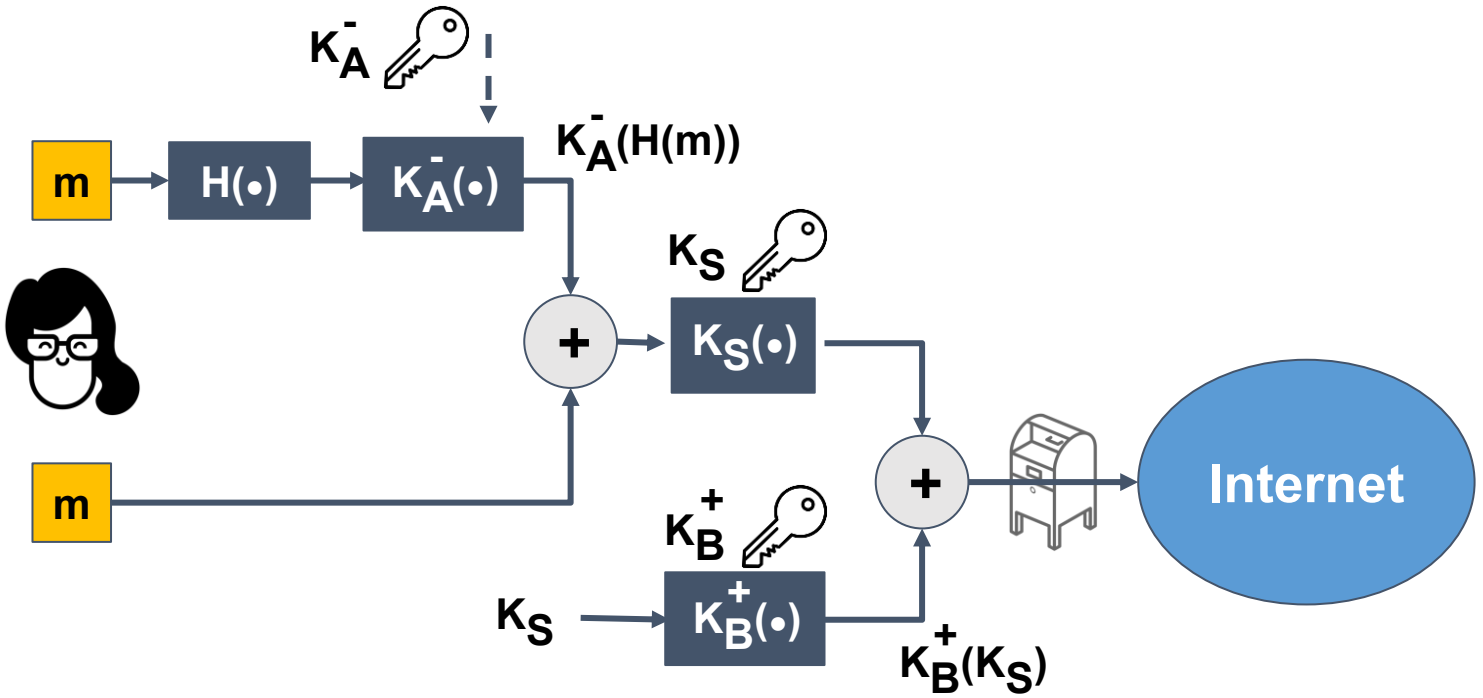
- Alice wants to provide sender authentication message integrity



- Alice digitally signs message
- Alice sends both the message and the digital signature

Everything Together

- Alice wants to provide secrecy, sender authentication, and message integrity



- **Alice uses three keys: Her private key, Bob's public key, and the newly created symmetric key**

Holy grail of everything learnt till now.

Pretty Good Privacy (PGP)

- **Internet e-mail encryption scheme, de-facto standard**
  - Uses symmetric key cryptography, public key cryptography, hash function, and digital signature as described
  - Provides secrecy, sender authentication, and integrity
  - Inventor, Phil Zimmerman, was a target of a 3-year federal investigation

A PGP signed message:

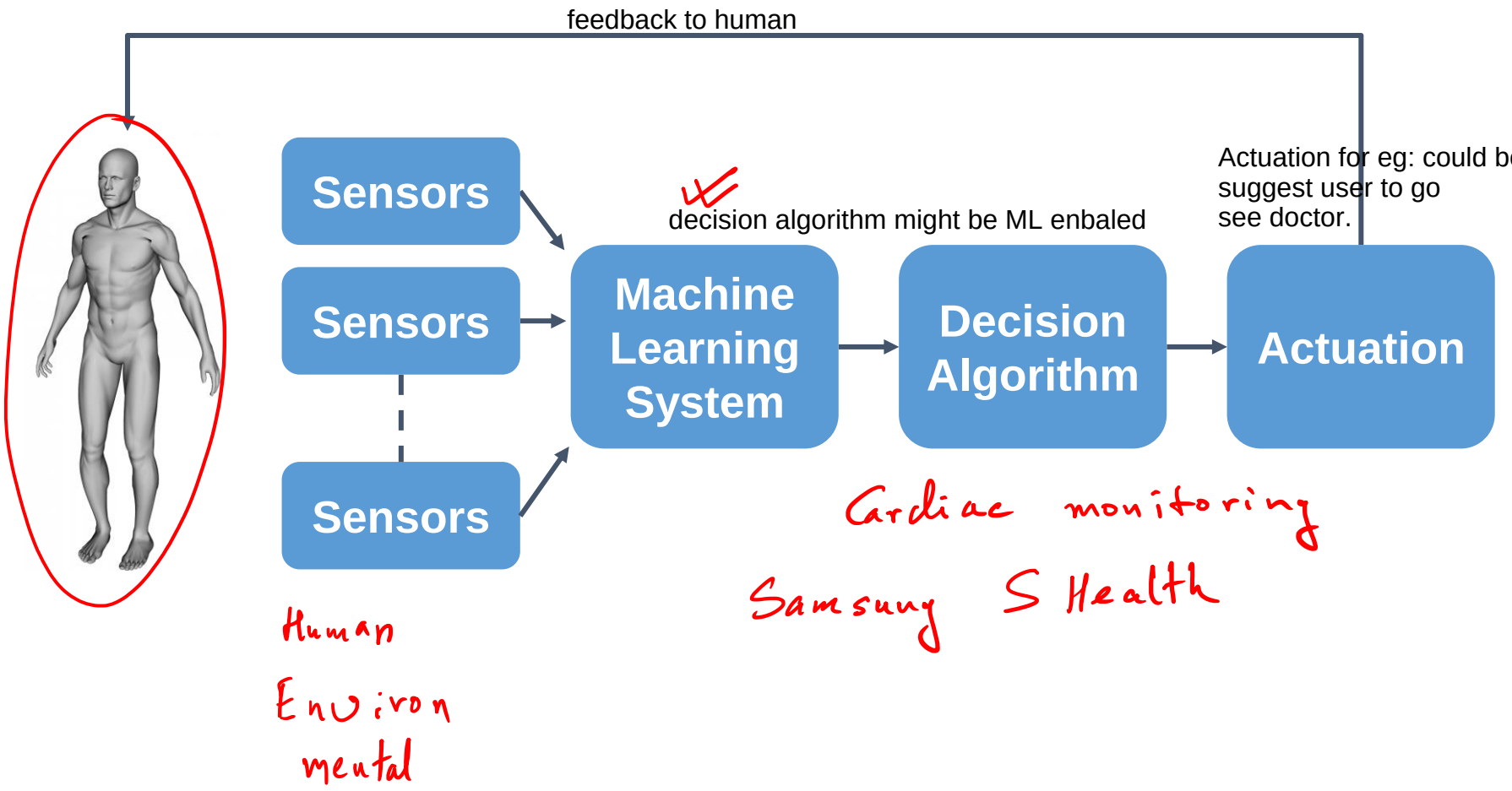
```
---BEGIN PGP SIGNED MESSAGE---
Hash: SHA1
Bob:Let's meet for a Latin
American dinner tonight. Best,
Alice
---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3m
qJhFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

Recent Trends in Mobile Security

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

System Model



Sensor can be on human body or in environment

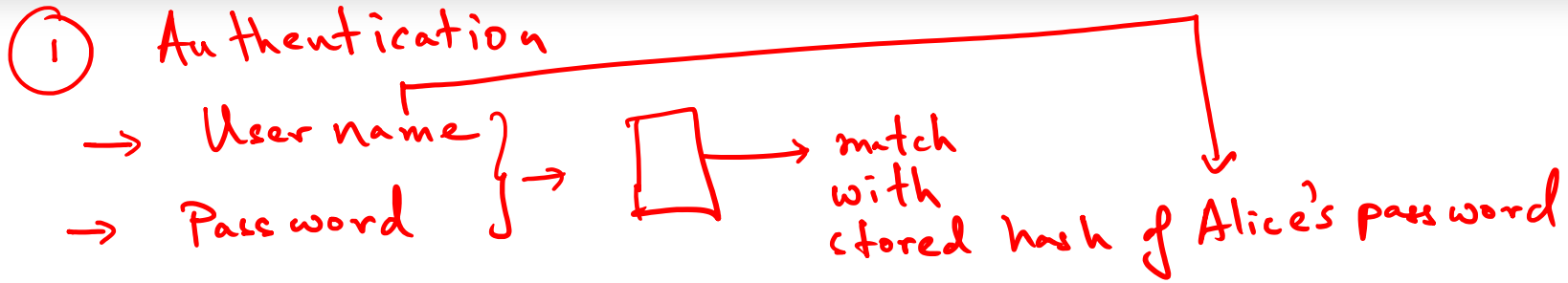
Security Problems

- Authentication
- Identification
- Data provenance
- Data integrity
- Data injection
- Data alteration
- Access Control

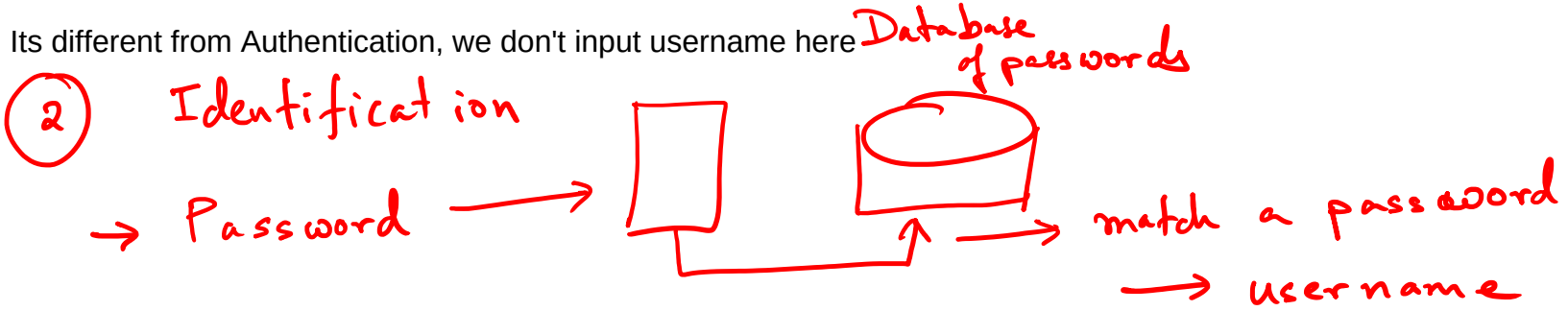
these are sub-categories of integrity

Authentication and Identification

Hsgdfhbow can server be sure that this message was send by Alice



Its different from Authentication, we don't input username here



Here output is username

Example: Bio metrics based identification

Facial recognition

We don't tell our username, it does self matching

Data Provenance

We generally assume this; the environment is not trying to fool the sensor

Assumption: Data source is typically not compromised

↳ falsified human augmentation

Example: Person → Type 2 diabetes

It's bcoz your are taking high sugar intake

↳ Arizona at 4pm MST → 117 F

> 50

doctor suggest, run 1 mile at 4 pm for every 3 candies you eat

1 mile ≈ 10 – 13 mins

Doctor → Fitbit → Steps as well as speed

Doctor prescribes fitbit because he know patient might be lazy.

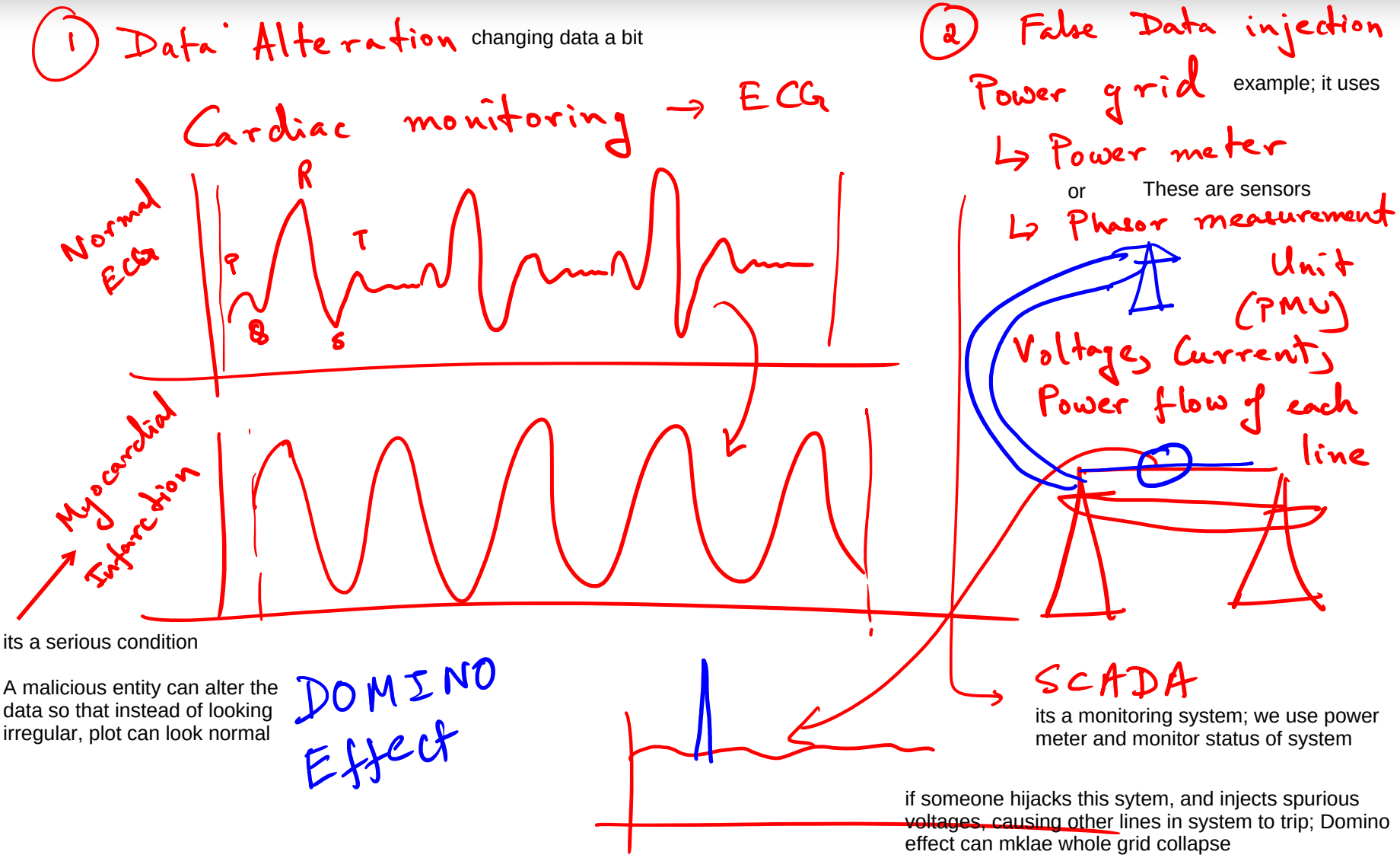
4pm → 4 mph

The goal of data provenance is: Ascertain that data is coming from correct source in the correct context

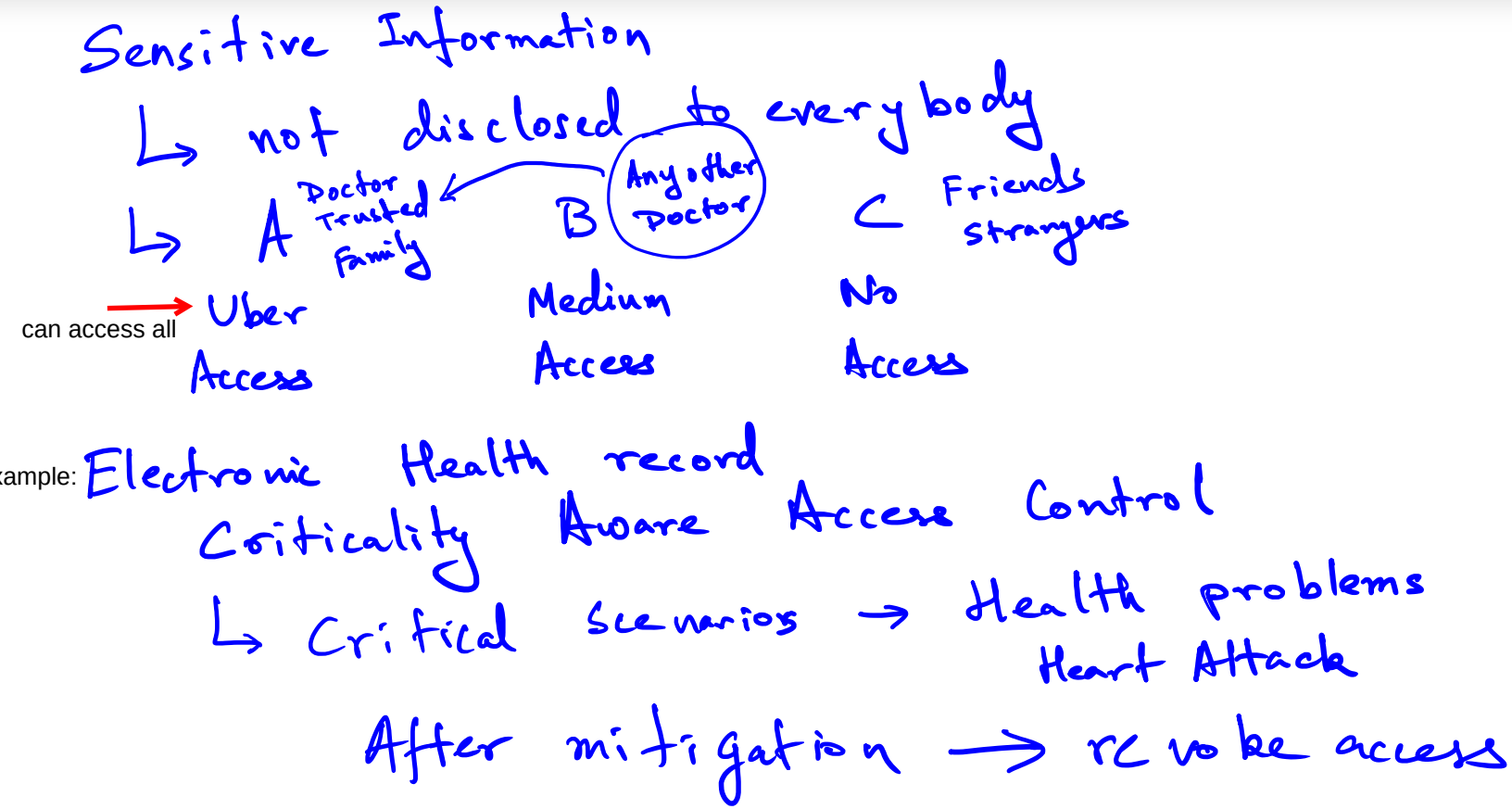
Person still eats candies, but instead gives fitbit to his son who runs at 4 pm, trying to foll the doctor

Data Injection

It has two components



Access Control



Idea is, in critical scenarios, we might give all access to even entity B or C
After all that is done, access must be revoked again

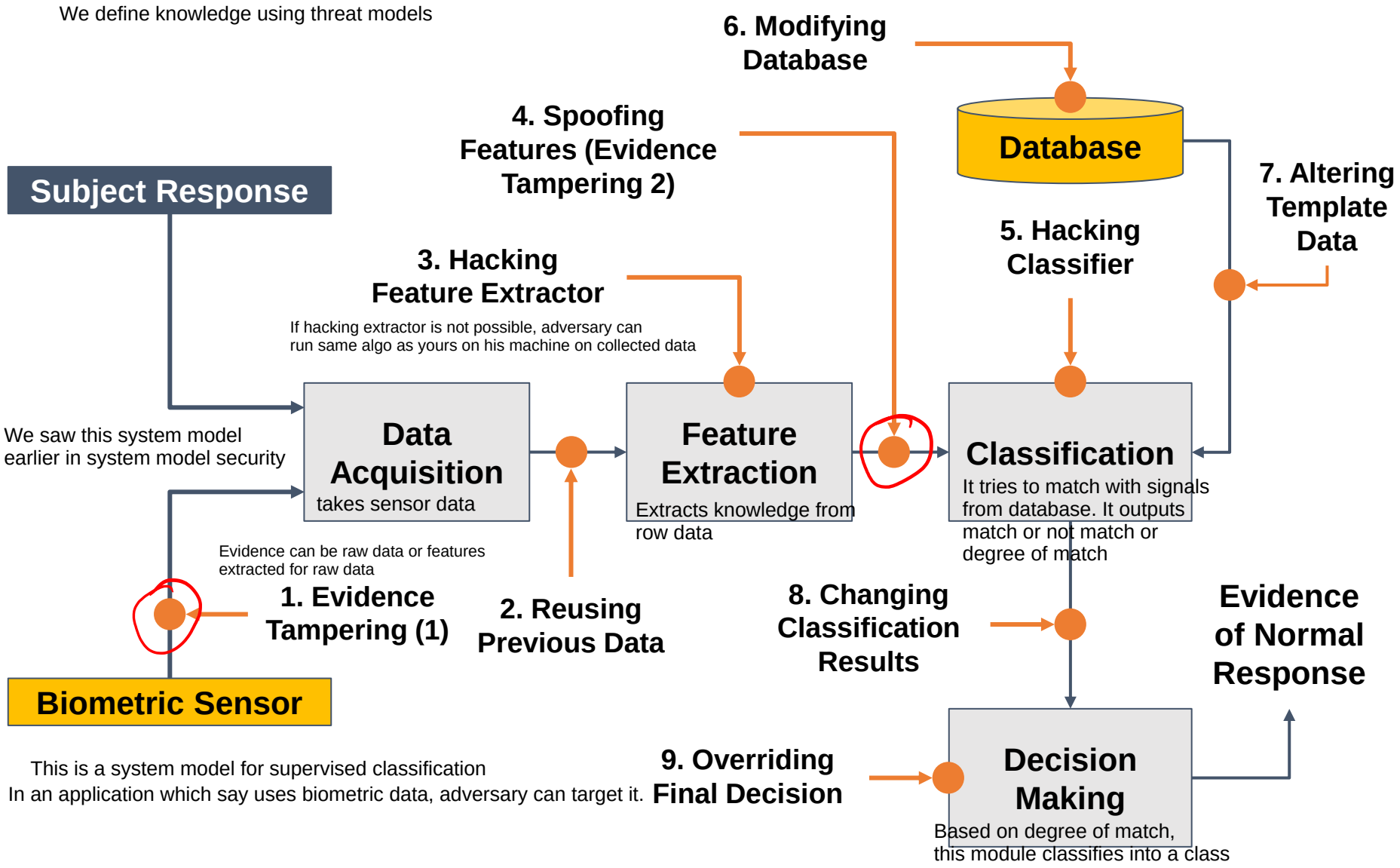
Mobile Security Threat Models

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Threat Model

Any security protocol need two things: Knowledge of adversary and capability
Knowledge means what adversary knows about the system
Capability means what adversary can do if he knows abot the system



Evidence Tampering

Access to evidence → knowledge

Capability is ability to generate false inputs

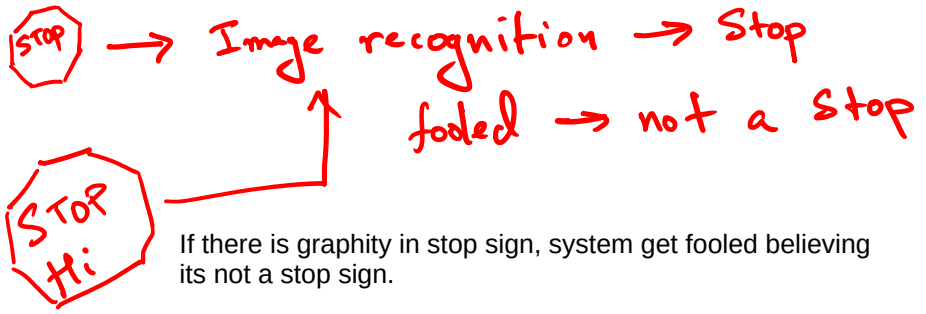
Without this you can't harm much

two ways of generating false input

Data injection

Data alteration

Example: Autonomous Cars



If there is graphity in stop sign, system get fooled believing its not a stop sign.

Cardiac Monitoring

Generative Model

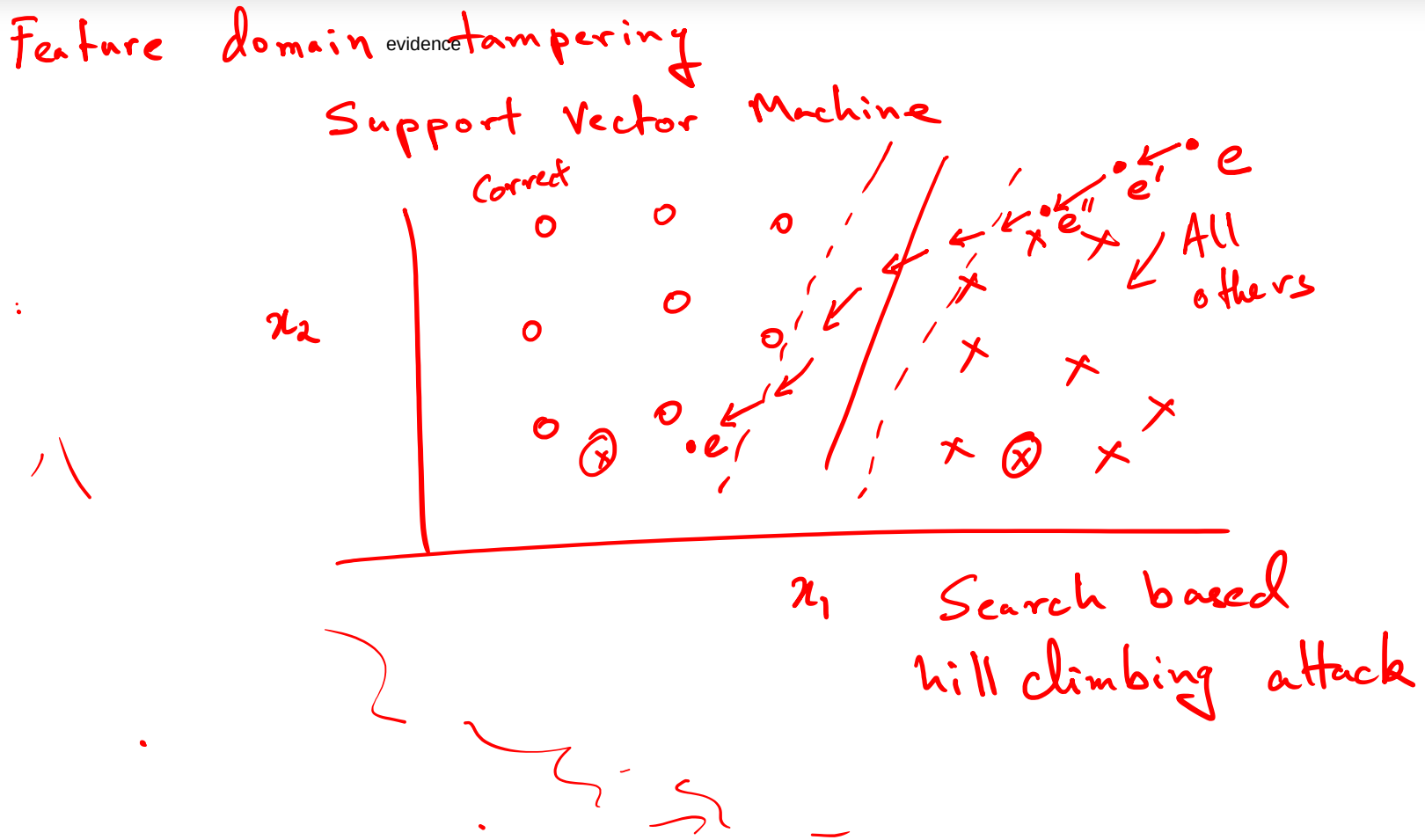
$$ECG(t) = \sum_{i=\{P,Q,R,S,T\}} a_i \exp \frac{(\theta-\theta_i)^2}{b_i^2}$$

$GM(a_i, b_i, \theta_i)$

15 params

Several ECG works learn generative model from ECG data where each wave is modelled as Gaussian. This is mathematical model of subject with 15 params. If we change any of the params, we get false ECG model which can be injected in ECG system

Evidence Tampering



say O is correct class of real user trying biometric authenictication. If we keep making small-small changes to X data points, we can get this to be classified as O (see arrows). Its called search based hill climbing attack.

Mobile Security Threat Models Poisoning

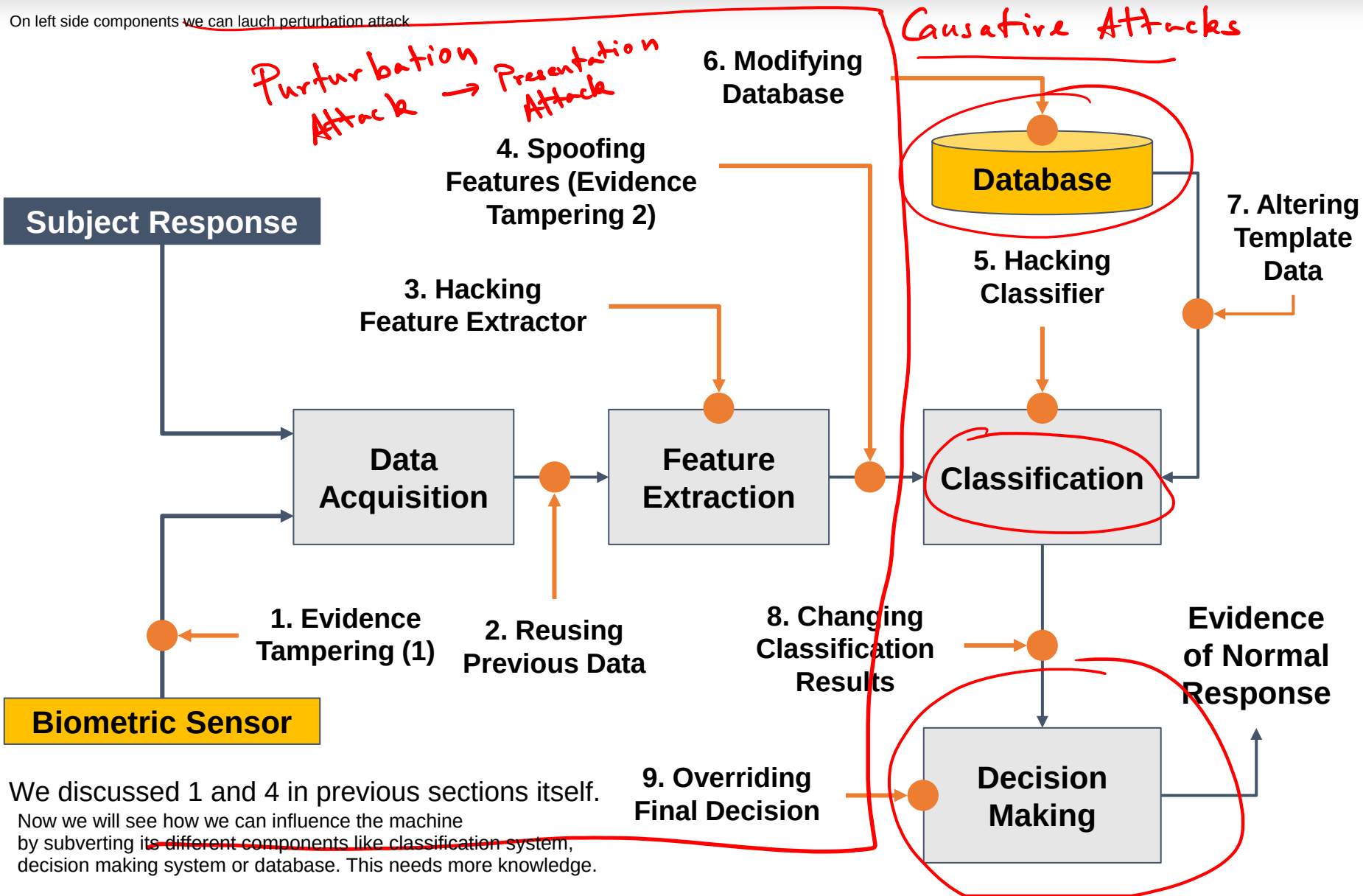
Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Threat Model

So there are two types of attacks possible on a MC system:
1. Perturbation/presentation attack deals with input of module
2. Causative attack: deals with parameters of decision making module; this needs more knowledge

On left side components we can lauch perturbation attack



We discussed 1 and 4 in previous sections itself.

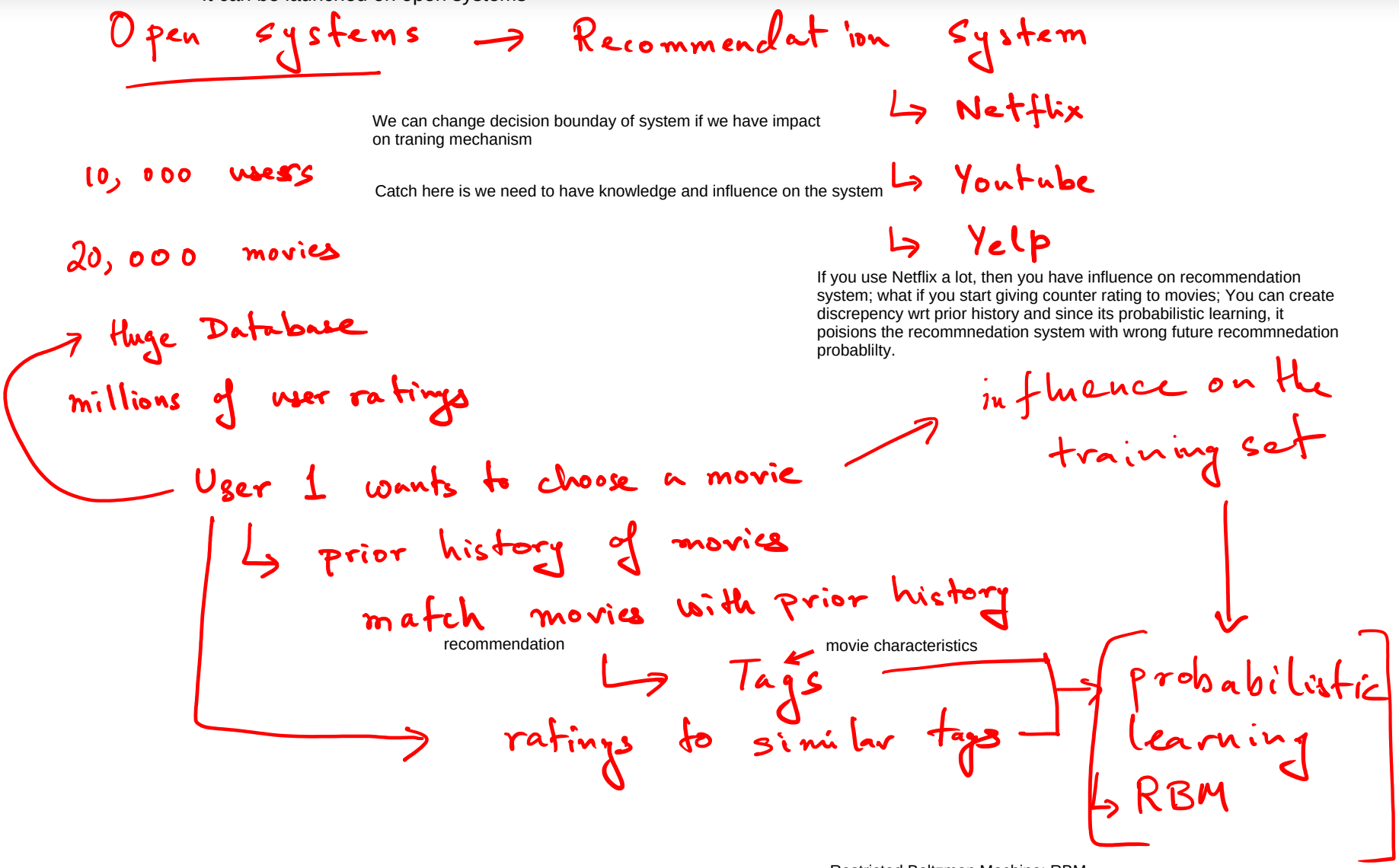
Now we will see how we can influence the machine by subverting its different components like classification system, decision making system or database. This needs more knowledge.

Poisoning Attacks

Apart from evidence tempering attack on MC system, poisoning is another type of attack that can be done

Poising is a causative attack (see in previous slide)

It can be launched on open systems



Restricted Boltzman Machine: RBM

Earlier we saw threat models
Now we will see how to access the knowledge required by such threat models

Mobile Security Knowledge Access

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

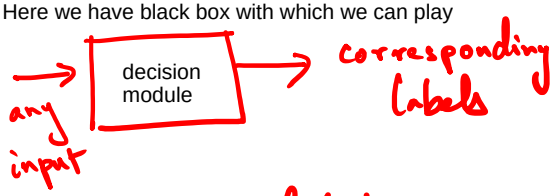
Ira A. Fulton Schools of Engineering
Arizona State University

Types of Knowledge Access

Three mechanisms of accessing knowledge:

Black Box

Here we have black box with which we can play



1.) Create a database
- ↳ large set of legitimate inputs
  - correspondin to those we will get
- ↳ correct labels output by black box

↓ Surrogate Data

Based on this data, we can

2.) Create a Decision Making Module (Data driven) mimics the Black box Surrogate Machine

Its like a proxy for black box module.
For those inputs, it will give same output as black box
If input data is large enough, we can mimic original system and harm their sales.

This machine running over surrogate data is called surrogate machine

So if adversary has white box access, a lot can happen.

White Box

1.) Machine type

2.) Internal params

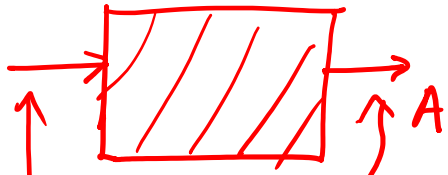
3.) Training algo

4.) Training Dataset

Here we know all internal details like Machine type, traning algo, internal params of machine and we also have access to traning dataset. What attacks we can do?
- One possible attack is Poisoning

Gray Box

This is somehwat more restrictive



cannot be any input of a specific class

if input $i \in$ Class A

if $i \in$ Class B get output once but system shuts down

if input i belongs to class B then it gives output once but shuts down after that so we can only get partial surrogate data; we have examples of only class A; with this we can't make good surrogate machine

This is what is practically used.

Mobile Security
Biometric Authentication

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Characteristics

Earlier Source of security was a difficult mathematical problem

$(g^a \bmod p) = x$

g = ↳ x

- **Usable security**
  - Least overhead, acceptable security
- **Idea: Use the human body for security**

This is a computationally difficult problem. So think of viable alternatives

- **What can be used?**

environment
↳ inherent randomness

We will discuss these 3 ways in details now

- Biometrics
- Physiological signals
- Activity and gestures

Biometrics (two basic characteristics of a signal to qualify for being called biometric)

↳ unique for an individual

↳ static → involuntary

Shouldn't change with time

These are not biometrics because they are voluntarily controllable

Transient Biometrics

1.) Static shape

Fingor prints

iris scan

2.) Varying time domain feature ECG

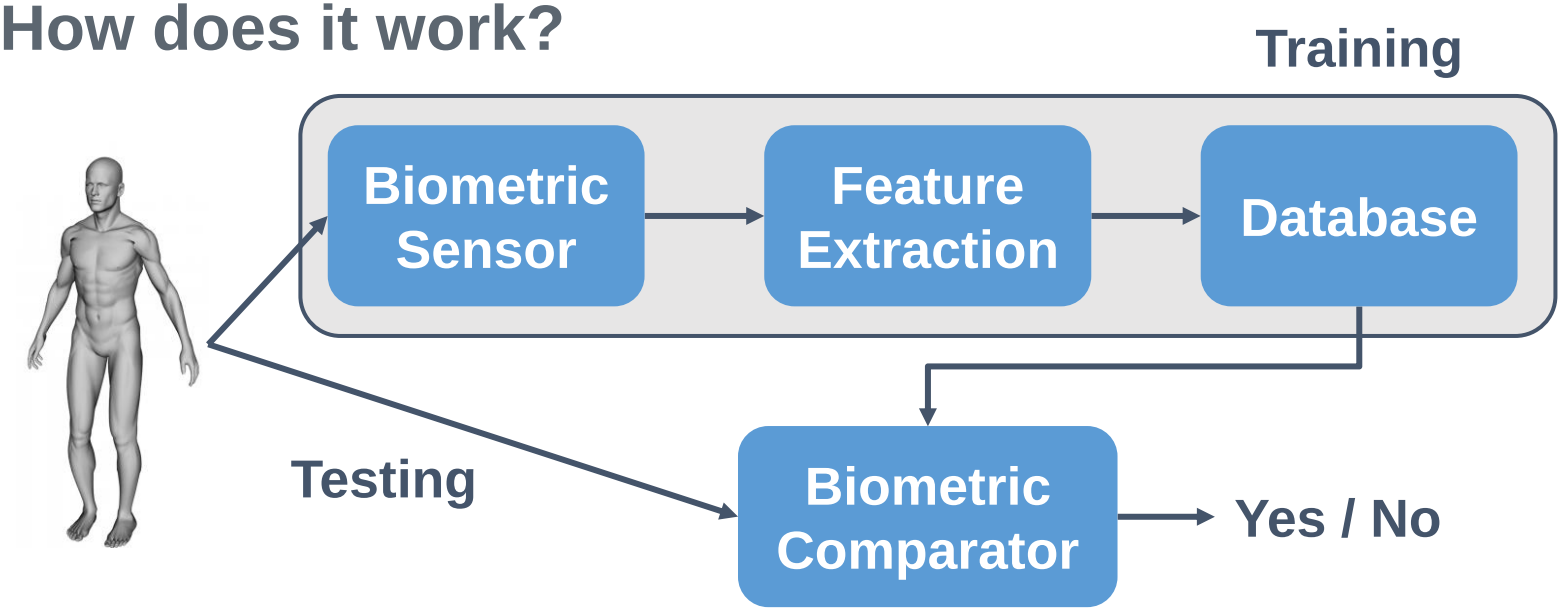
Problem is once biometric is stolen, real user can't use them ever again.

ECG shape doesn't change for healthy people

Some propose changable biometric - they use random number hashing or xor etc. But that has its own problems.
That is where notion of transient biometric came into picture

Biometrics

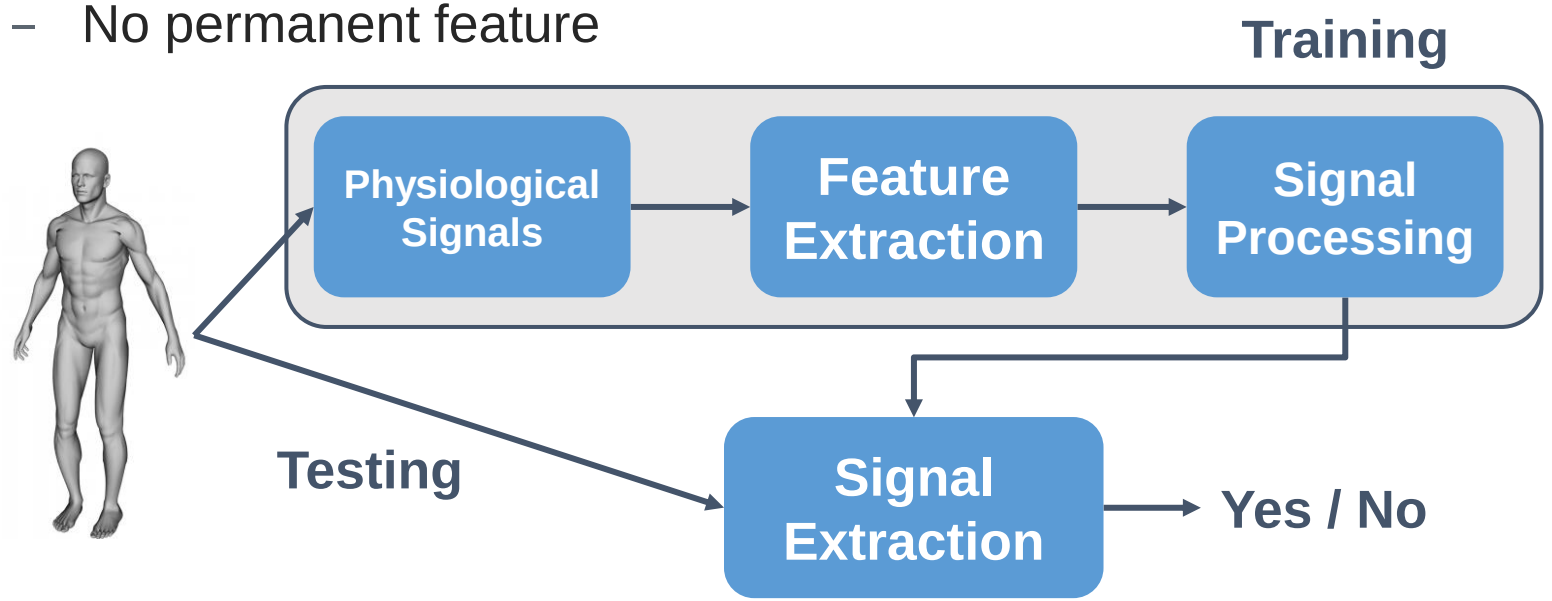
- **Some form of unchanging and personal information**
  - Examples:
    - Fingerprints
    - Iris scan
    - Facial scan
- **How does it work?**



Physiological Signals

- **Some form of signals obtained from the human body are in free living conditions without intervention from the human user**
- **How is it different from biometrics?**
  - Time varying
  - Highly random
  - No permanent feature

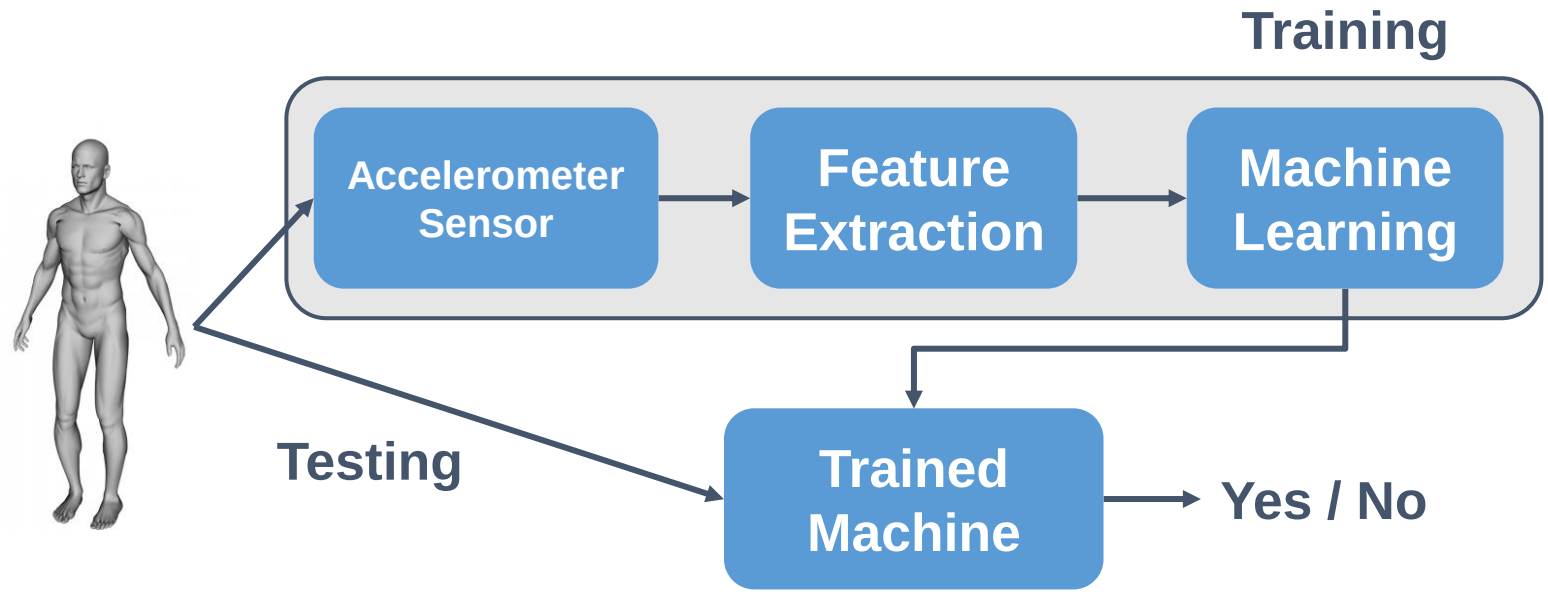
We need to sepnd lot of time processing signals



Activity and Gesture

- Depends on personal habits
- Human computer interface

Here we have no understandiong of how signals should look like.
So we can noway design a deterministic algorithm. That's why we use ML here



PSKA

Its similar to Diffie Hellman which can't be implemented on sensors because of overhead; those problems are not there in PSKA

= Diffie Hellman

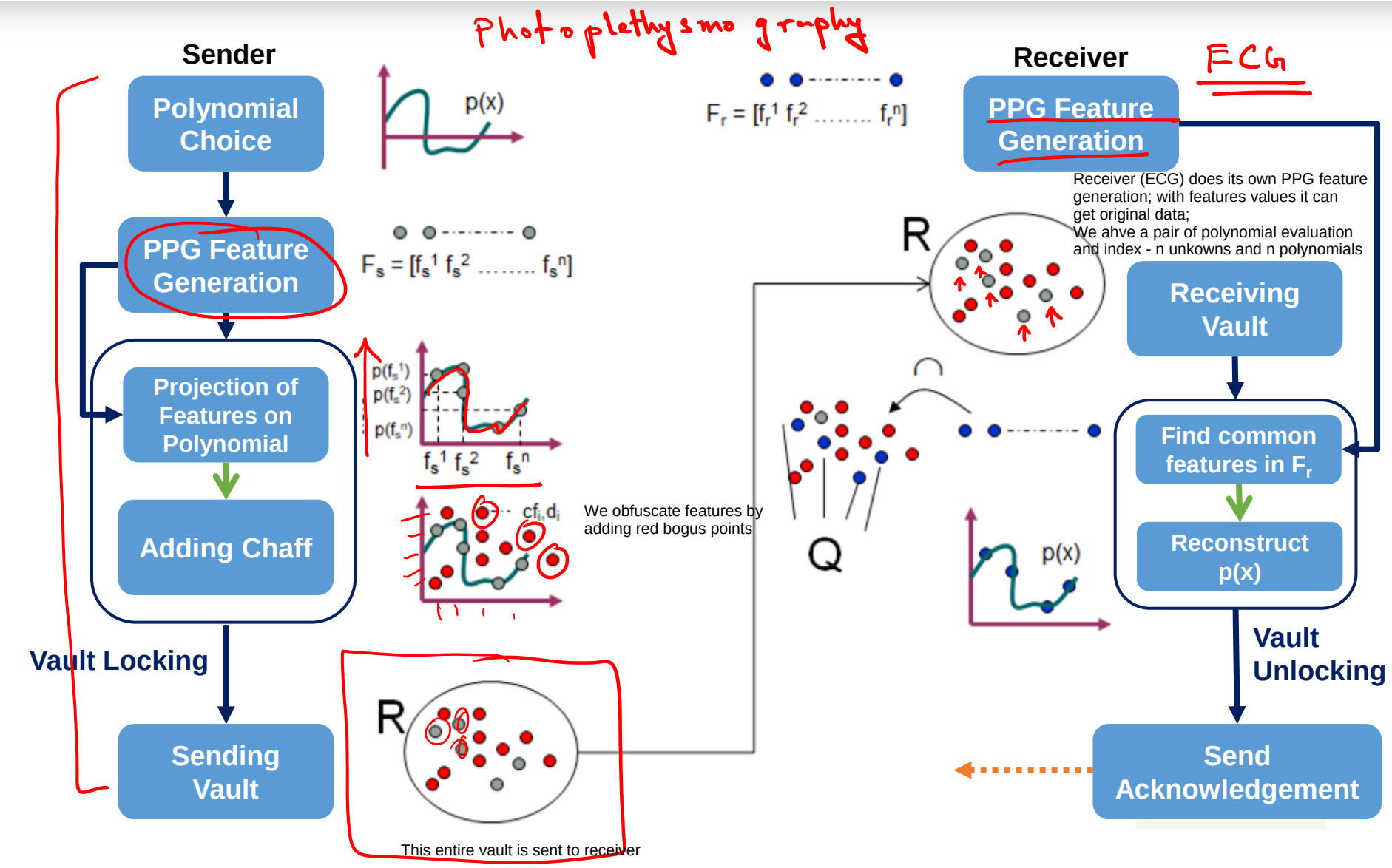
With PSKA, PPG pulse rate and heart rate sensors can securely communicate without sharing a secret, human body is secret

Both bloos pressure and ECG sensor can measure heart rate,
From heart rate, we can extract complex feature (PPG part) which are unique for everyone
These features vary with time because we are using transient biometrics

Physeological signal based Key Agreement protocol

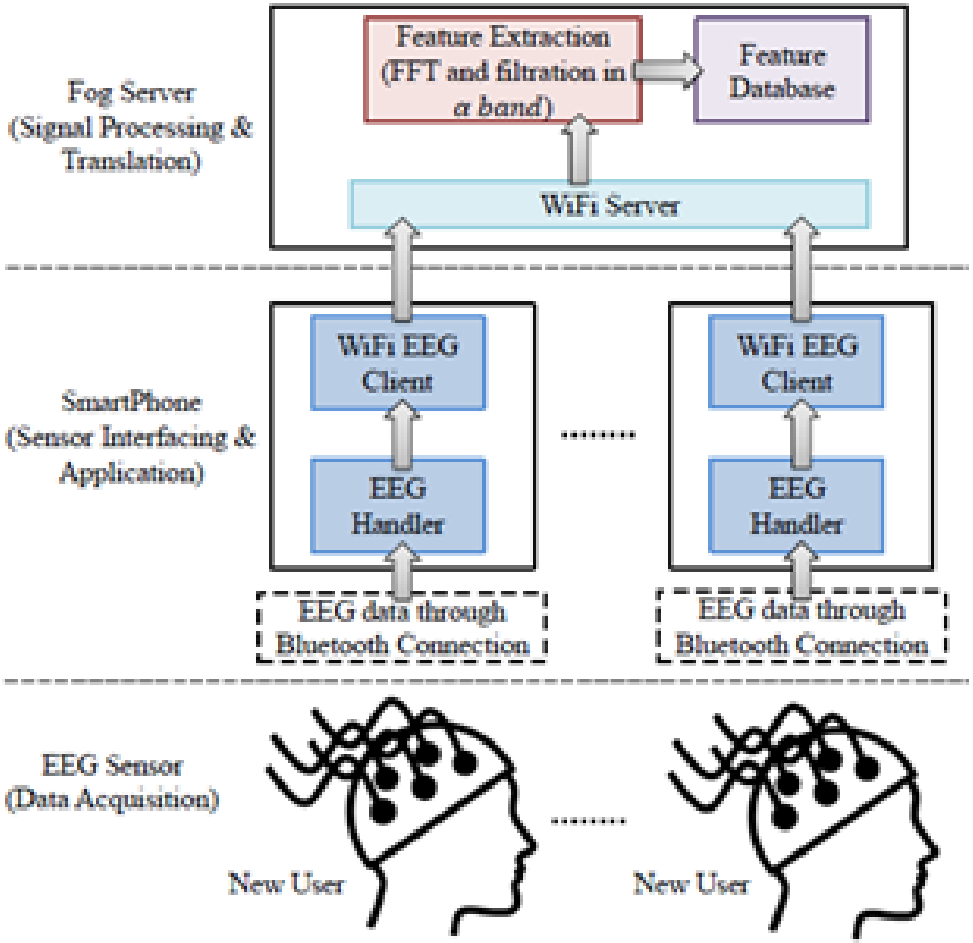
Photoplethysmography

ECG



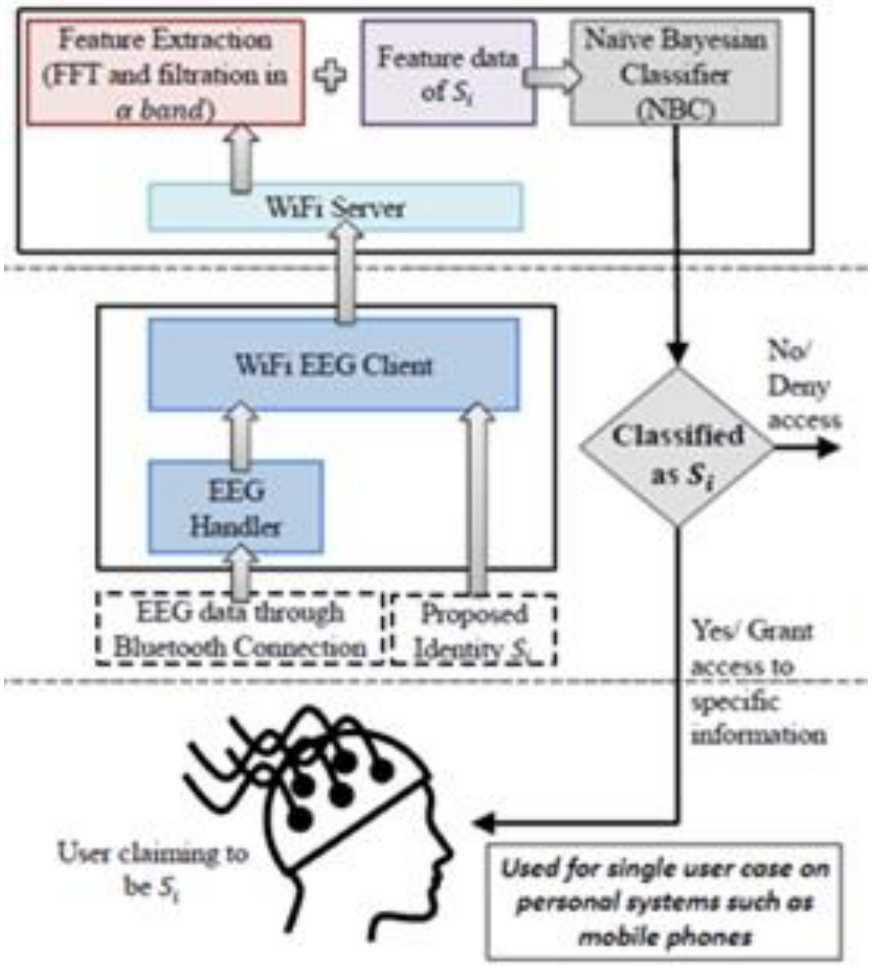
EEG Based Authentication

We can use brain data for EEG based authentication and identification. No signal processing possible. Only ML is the way.



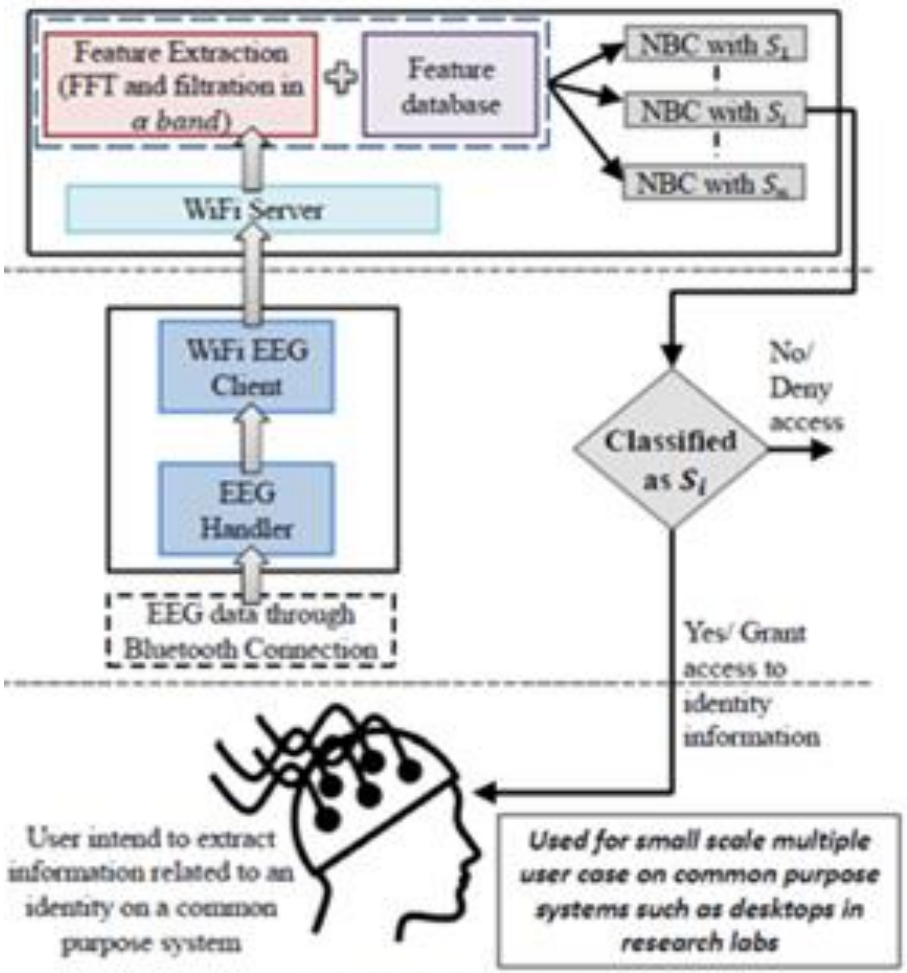
a) REGISTRATION

EEG Based Authentication



b) AUTHENTICATION

EEG Based Authentication



c) IDENTIFICATION

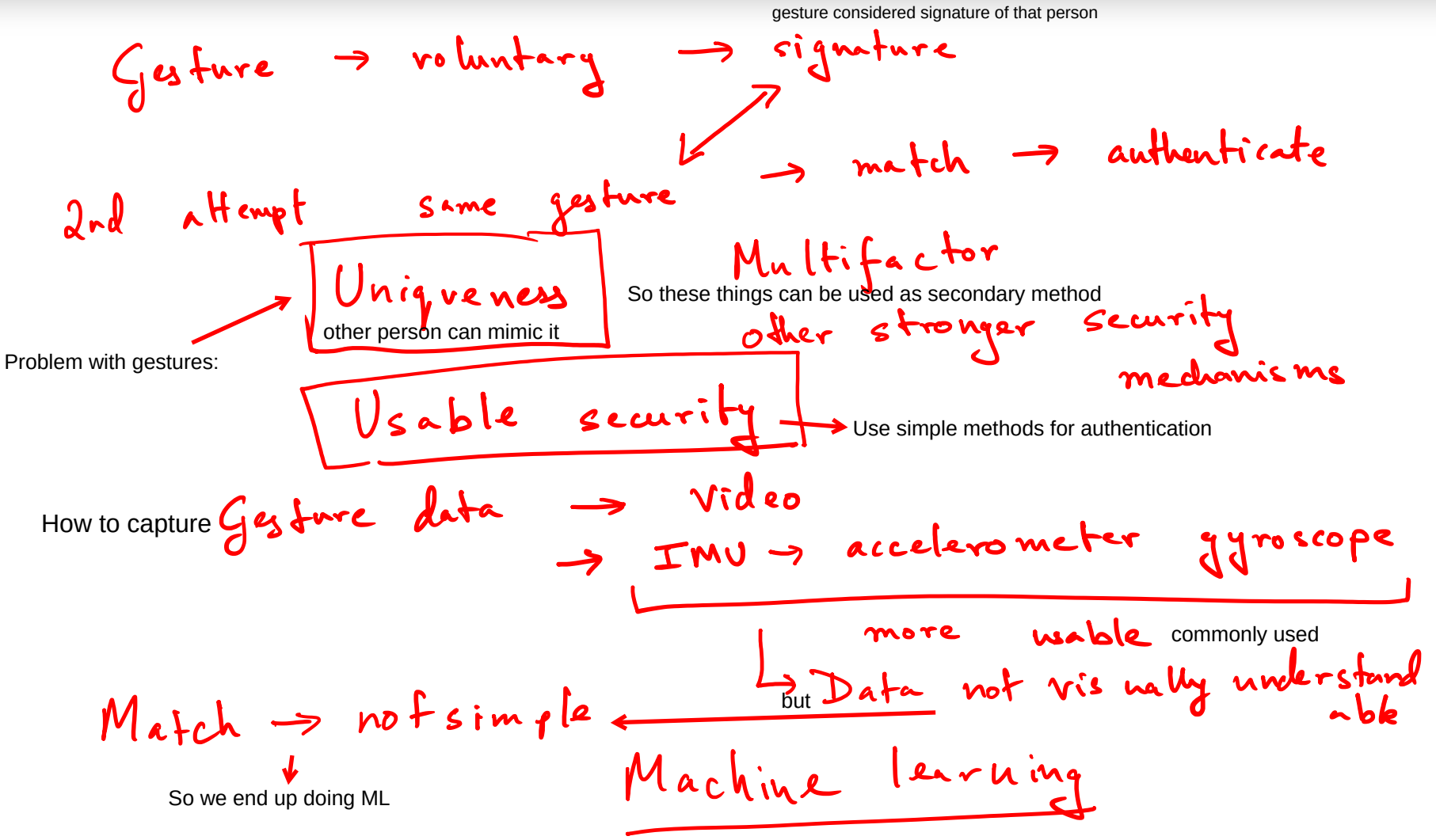
Adversarial Machine Learning

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Adversarial we here are discussing here in terms of security

Ira A. Fulton Schools of Engineering
Arizona State University

Gesture and Activity based Security



There are heterogenous sensors, simple algo based amtching is difficult

Now adversary has to fool ML system. That's where adversarial ML comes into picture. Shifting focus on performance of ML system when under attack

Machine Learning Security

- **What is the effort of an adversary to fool a machine?**
- **Many ways to fool a machine**
  - Brute force feature guessing
  - Generate a signal that looks similar
  - Evasion attack (false data trying to get classified into wrong class)
  - Poisoning attack affect traning of ML ssytem such that it gives wrong results. eg: Netflix recommendation system

it depends on knowledge and capability of adversary

There is a tug of war - arm's race between adversary and ML system

Adversarial ML: Arm's Race

Knowledge

Process Model ⇒ Threat Model

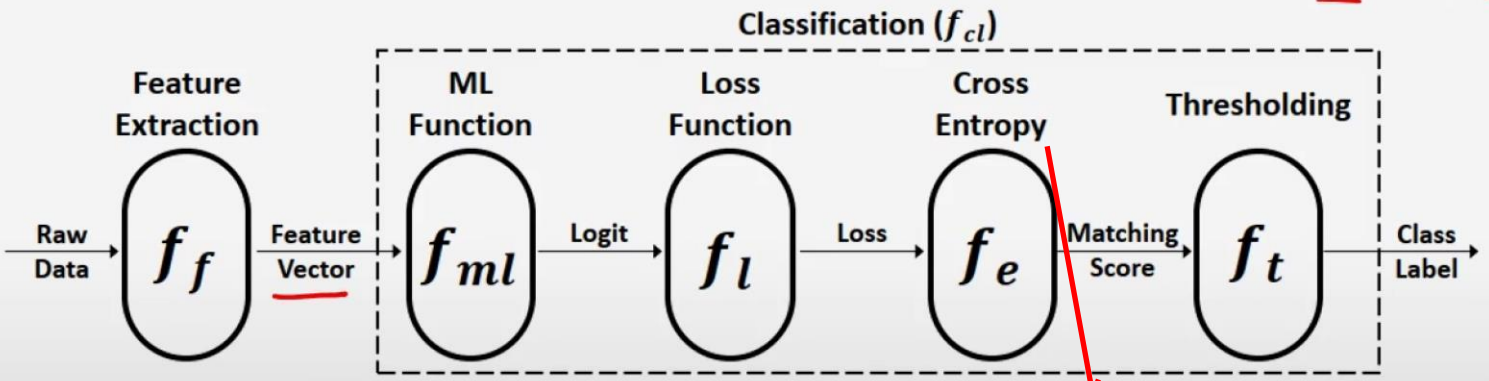
process model/threat model of any ML system

A → $m_A > Th$!

B → m_B > Th ? $m_x > Th$ then class label → X

C → m_C

D → m_D



SVM → $\vec{w} \cdot \vec{x} + b$ → logit

ML function can talk many forms dependeing on Model used

Loss function can also take many forms

$\exp\left(\frac{\text{logit}^2}{\sigma^2}\right)$

$\frac{1}{1-\exp\left(\frac{\text{logit}^2}{\sigma^2}\right)}$

How good of a match is feature x with a representative from Class A?

Example is Euclidean distance, bayesian probability score etc

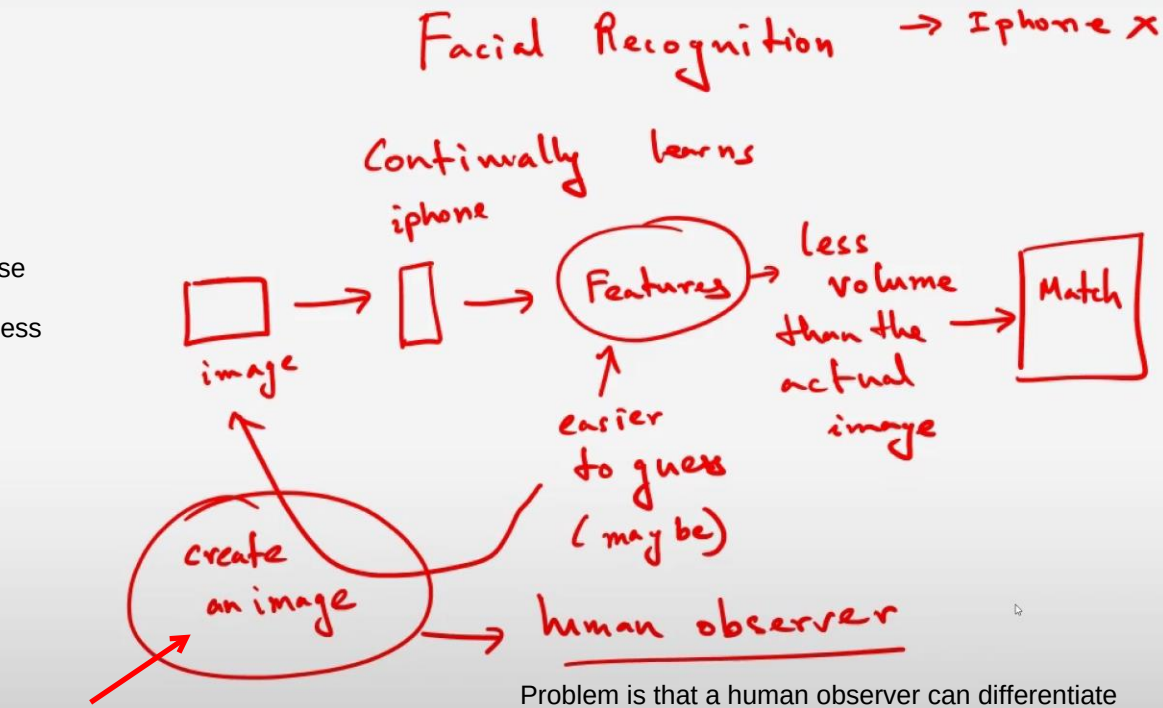
Capability

Black Box

White Box

gray Box

Adversary can access any of these knowledge bits.
Earlier we saw three ways to access knowledge - black, white, grey.



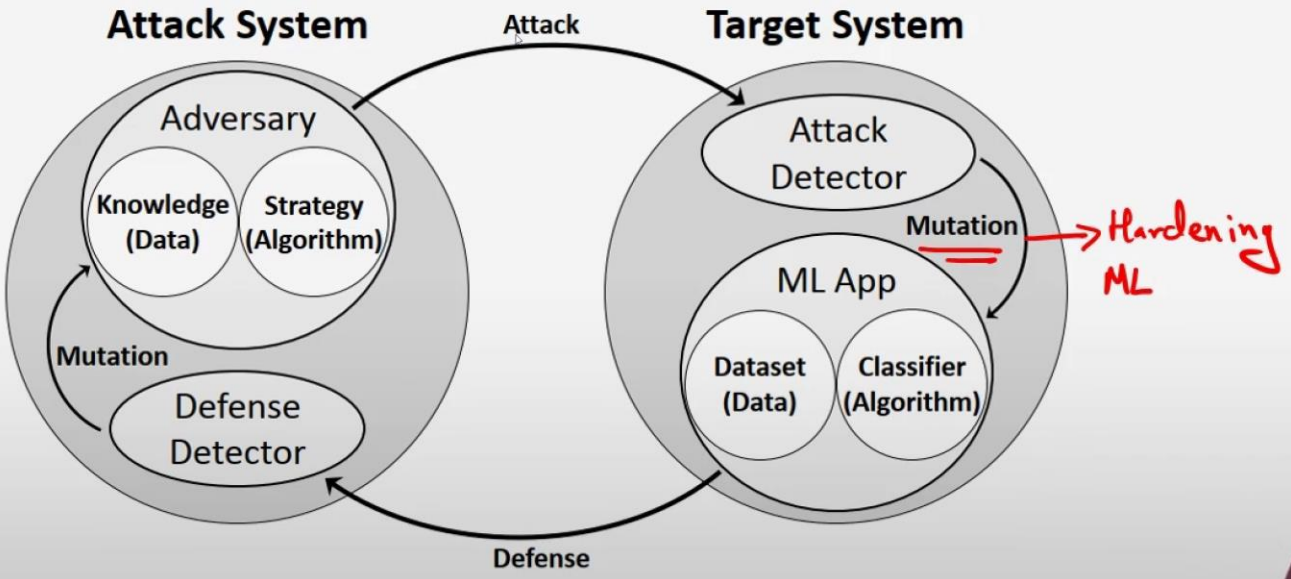
Problem is that a human observer can differentiate

adversaray can try this → So this is not easy

This is where capability comes into picture
You may have knowledge but not capability to fool the system

Arm's Race

Based on this knowledge and capaility an attack ssytem can be designed



On detecting attack, target ML system will mutate
Attack system sees that defence and it also mutates
This continuous mutation is called adversarial cycle - arm's race. They both are comparing
It mutation/hardening improves both target and attack sytems.

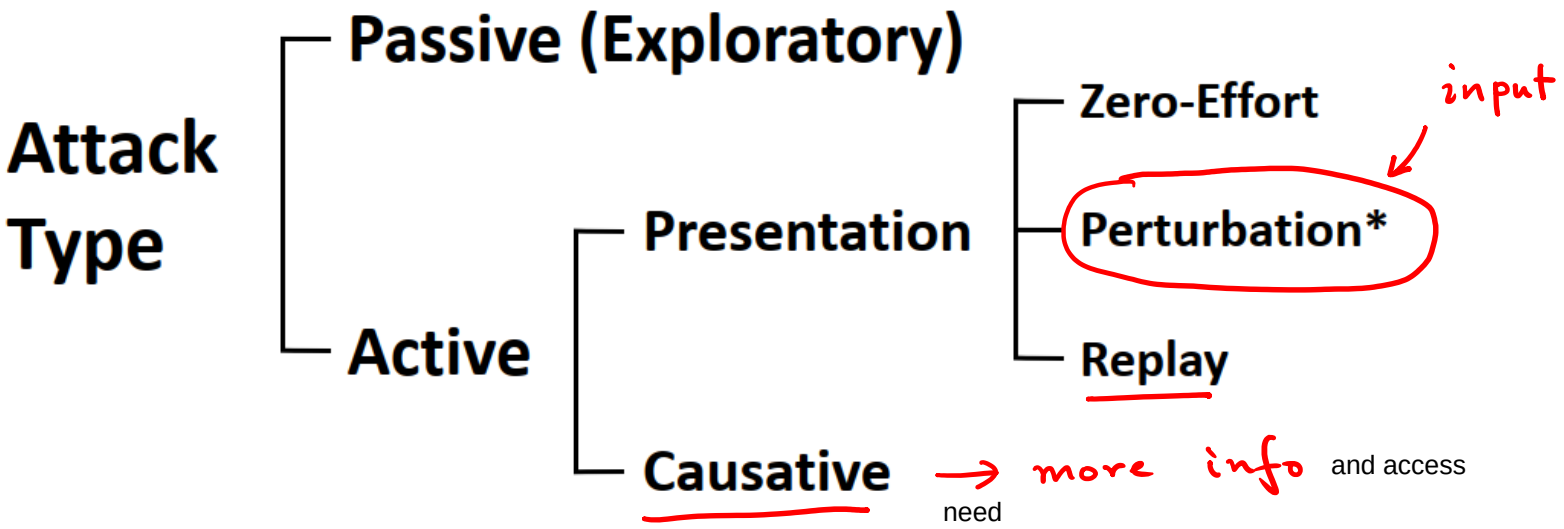
Adversarial ML Attack Types

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Attack Types

Direct function of knowledge and capability of attacker



Passive or Exploratory

Launch eavesdropping attack

⇒ main aim is gain knowledge about the machine

Surrogate Datasets ⟶ Surrogate machine

(if we have black box acces, we can make surrogate machine)

Recommendation System → Restricted Boltzman Machine (RBM)

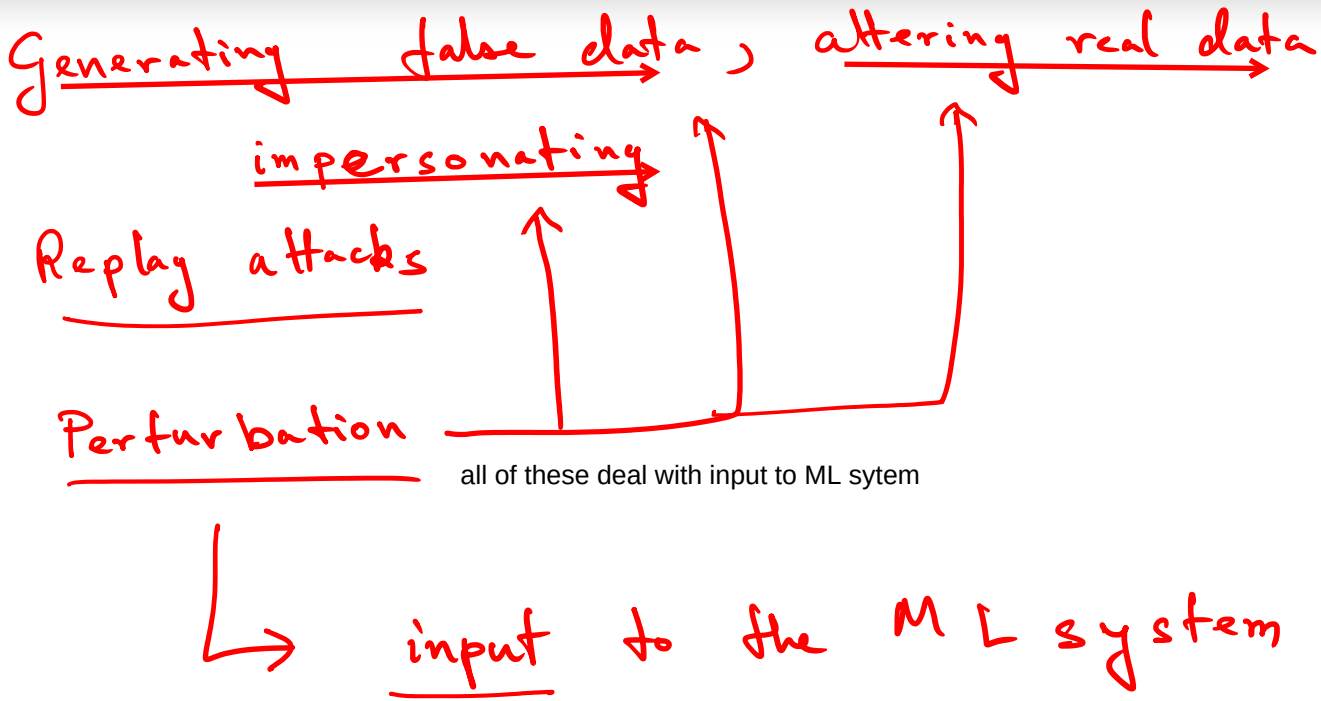
Launch an attack (Active) on the surrogate machine

Then we launch active attack on surrogate machine
Assumpotion is a successful attack on surrogate machine will succeed on real machine too (not mandatory)

Successful attack on surrogate

⇒ on real machine

Presentation



Causative

More difficult to launch because causative attack imply you know more about machine

Poisoning Attack

Subverting class labels

Database ← inject false data

⇒ Greater access to the machine

Open systems → Recommendation

Online training application

Such systems are prone to causative attacks

For now, we will focus on perturbation attacks only

Adversarial ML Perturbation Attack

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

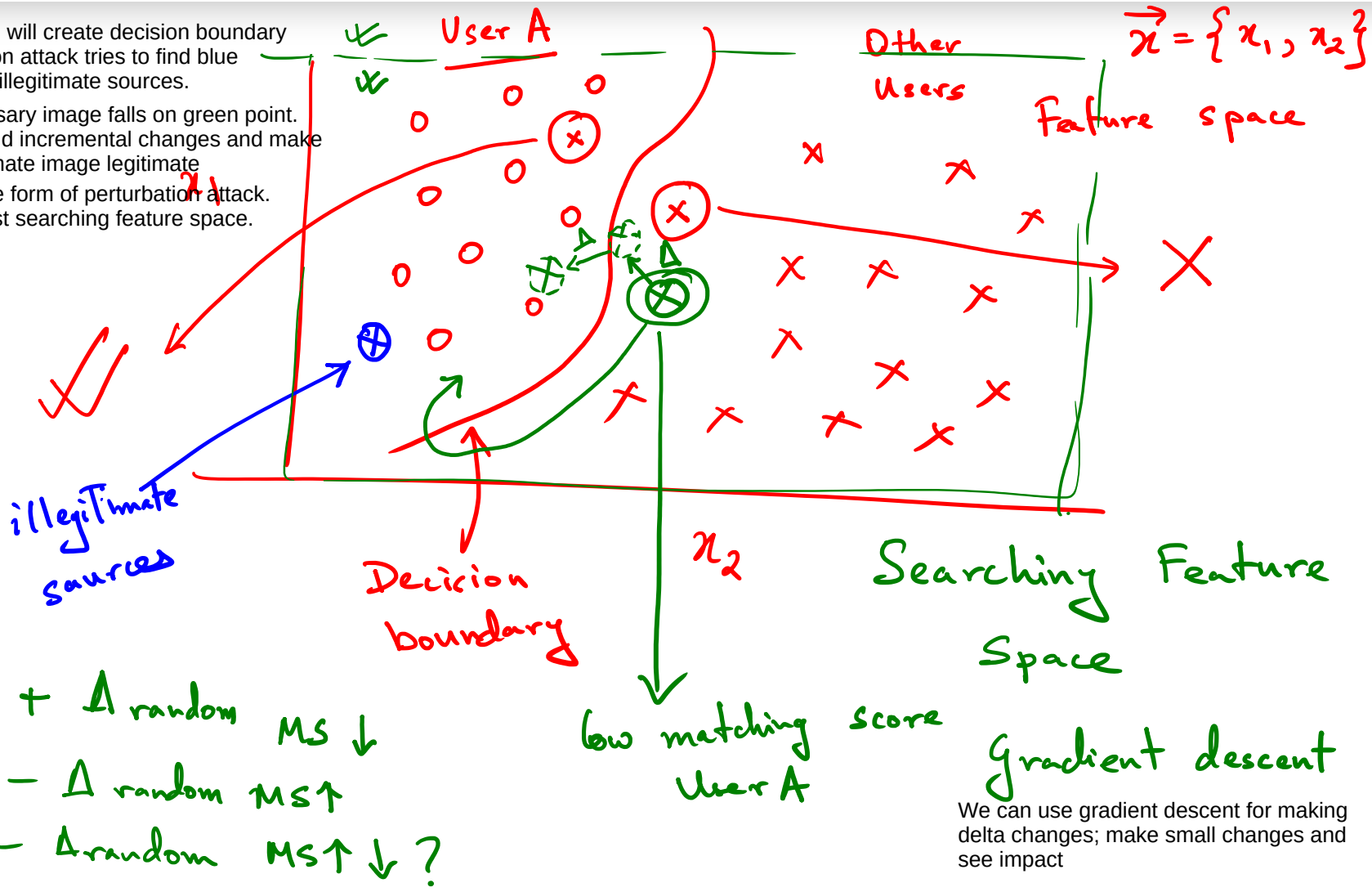
Perturbation Attack

Means we have to develop some input to subvert the machine

ML system will create decision boundary
Perturbation attack tries to find blue point from illegitimate sources.

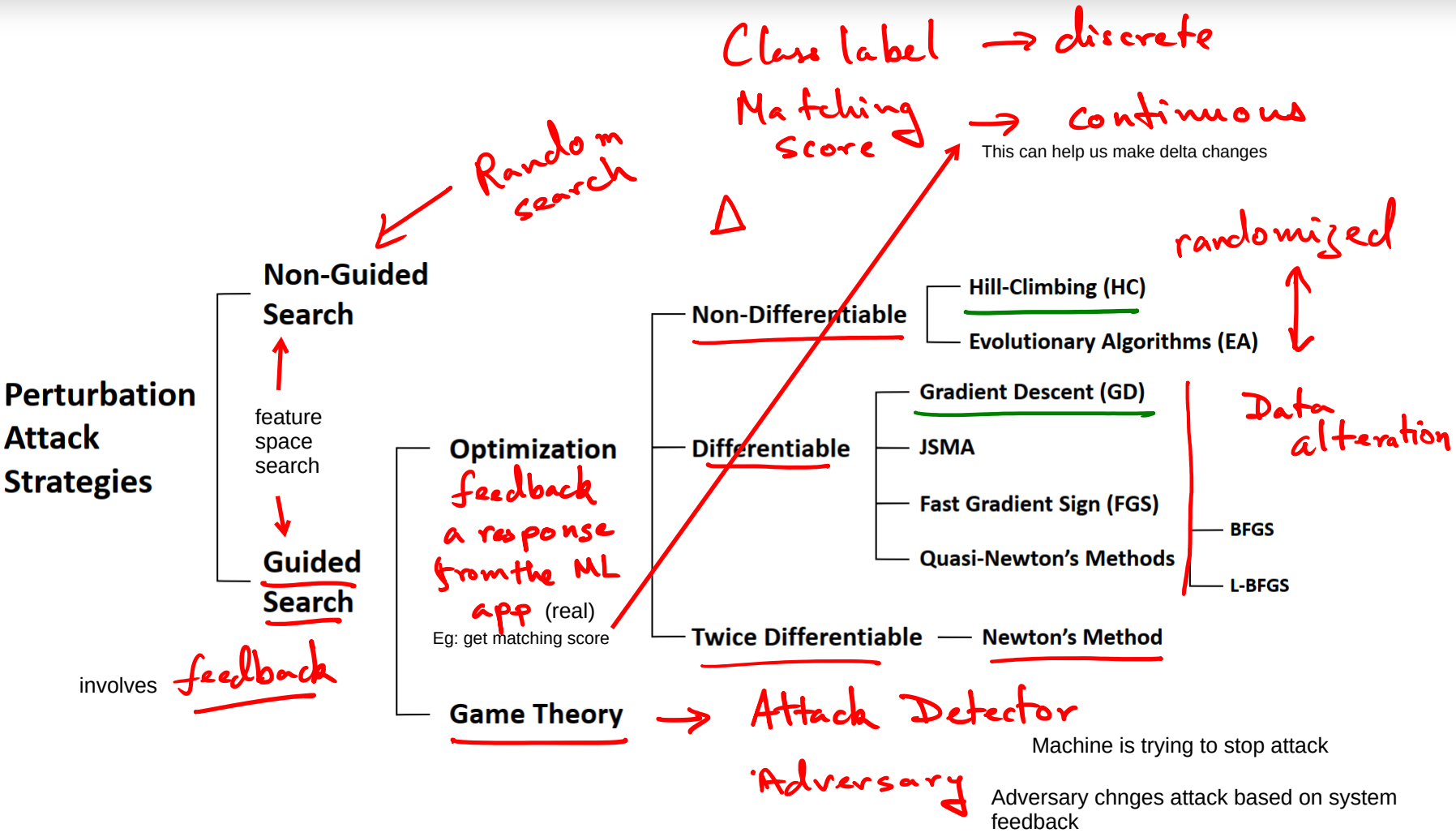
Say adversary image falls on green point.
We can add incremental changes and make my illegitimate image legitimate

This is one form of perturbation attack.
We are just searching feature space.



We can use gradient descent for making delta changes; make small changes and see impact

Perturbation Attack Types

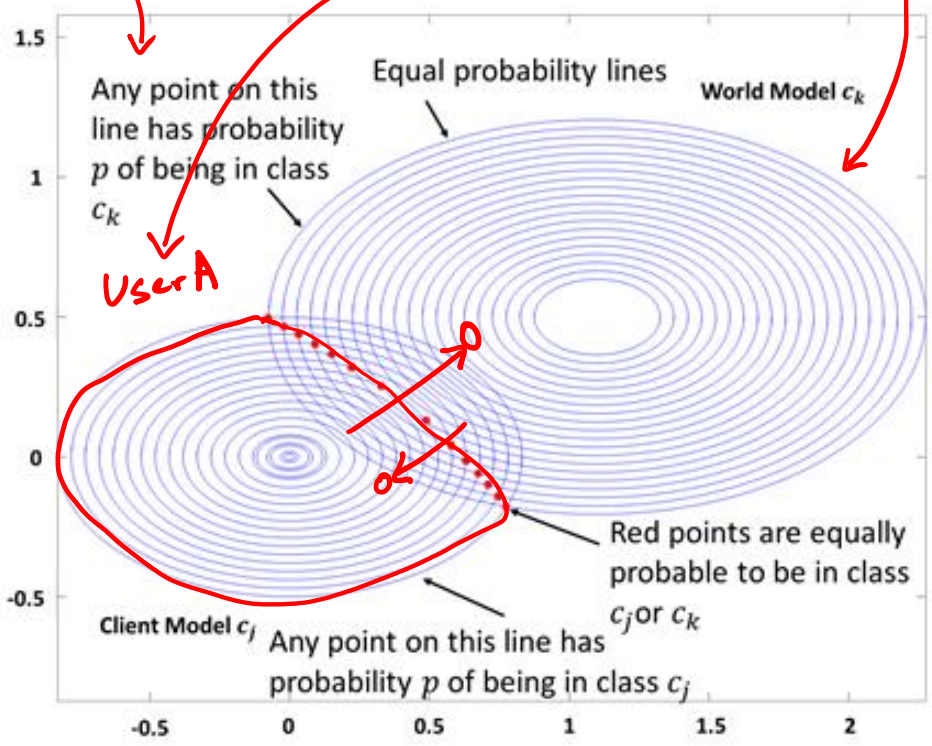


Brute Force

Examples of feature space searching

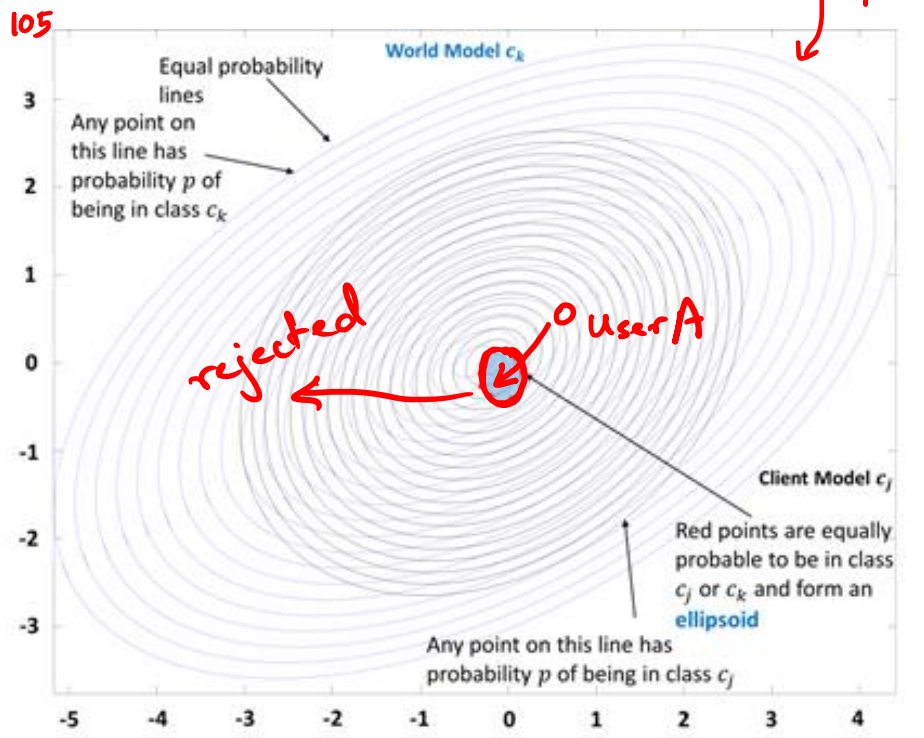
Example for Naïve Bayesian Classifier

EEG based authentication



Red dots are decision boundary
red circles are user A, other circles are other users

In this configuration, A still has large area/probability



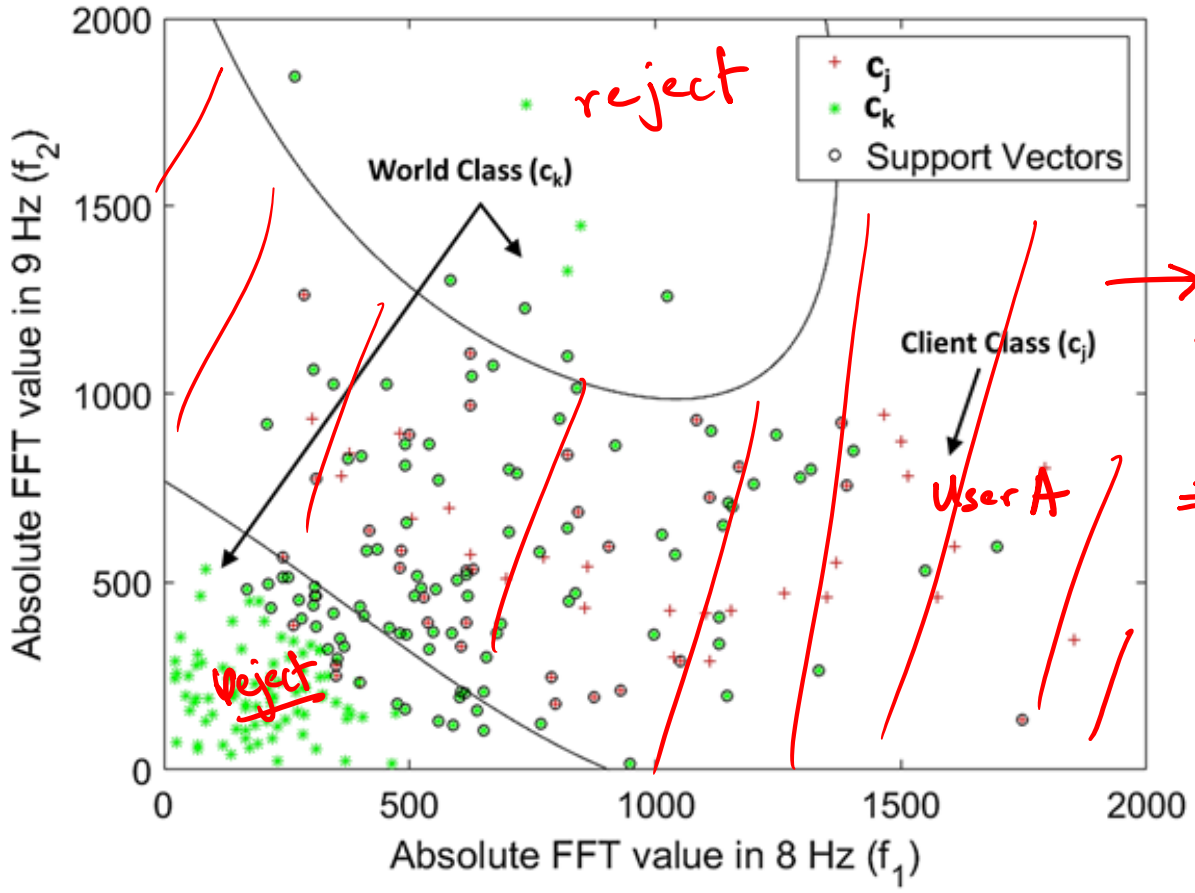
more difficult

In this configureation, we need to find point in event smaller area.
So for same machine, by changing configuration of machine, we can make attack difficult

Brute Force 2: Evasion

Example support vector machines

Red lined part is classified as A. So SVM has better accuracy but it also is more prone to be attacked when compared to Naive Bayes classifier above



So security depends on type of machine (SVM vs Naive Bayes) as well as parameter/configuration of the machine (previous slide)

Generative Models → false data generation

- Generate a signal that is similar (not the same) to the original
- Heart signals and brain signals can be modeled mathematically
- Parameters can be learned if data is available
- Can generative models fool machines?
- Usage in evasion attacks

Also called - any perturbation attack can have this

Adversarial Sample
Manufacturing
Data generation
Data Alteration

Adversarial ML Causative Attack

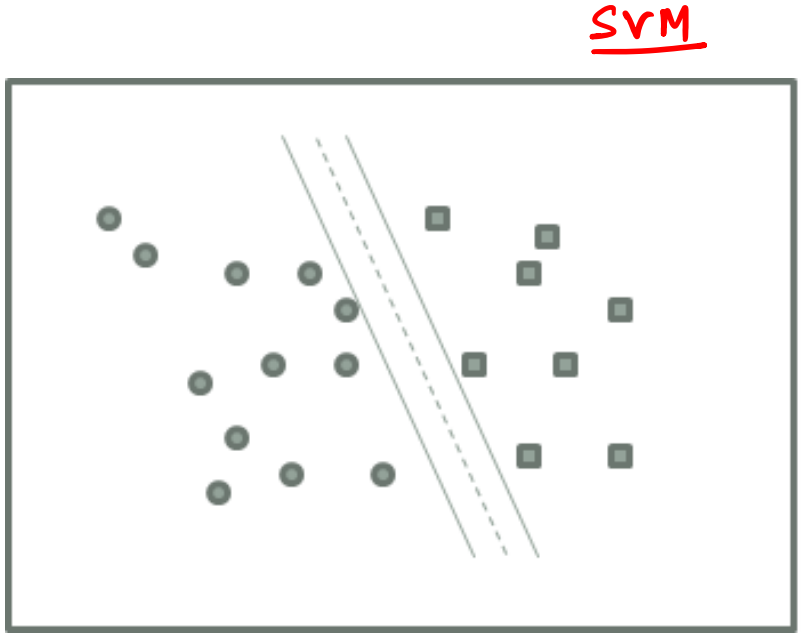
Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

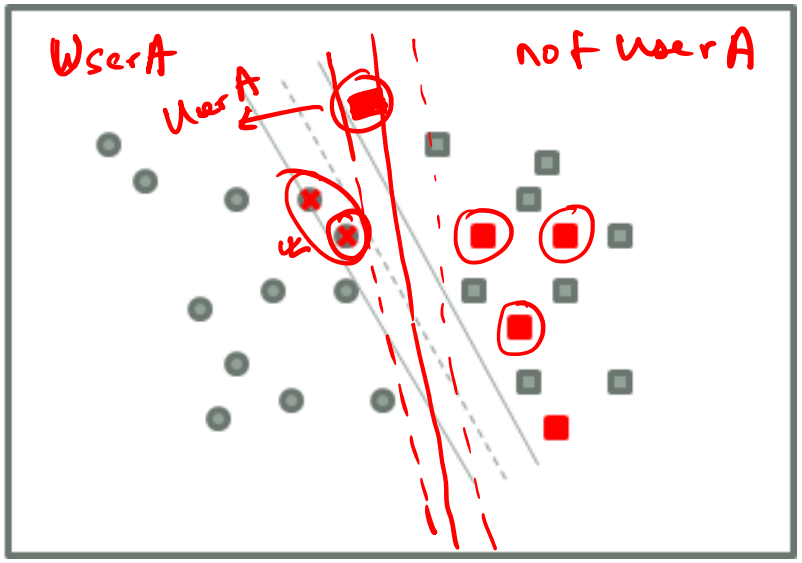
This needs more info and also more capability
eg: Poisoning

Ira A. Fulton Schools of Engineering
Arizona State University

Poisoning

- Access to training data
- Can we infuse training data to change a decision boundary?



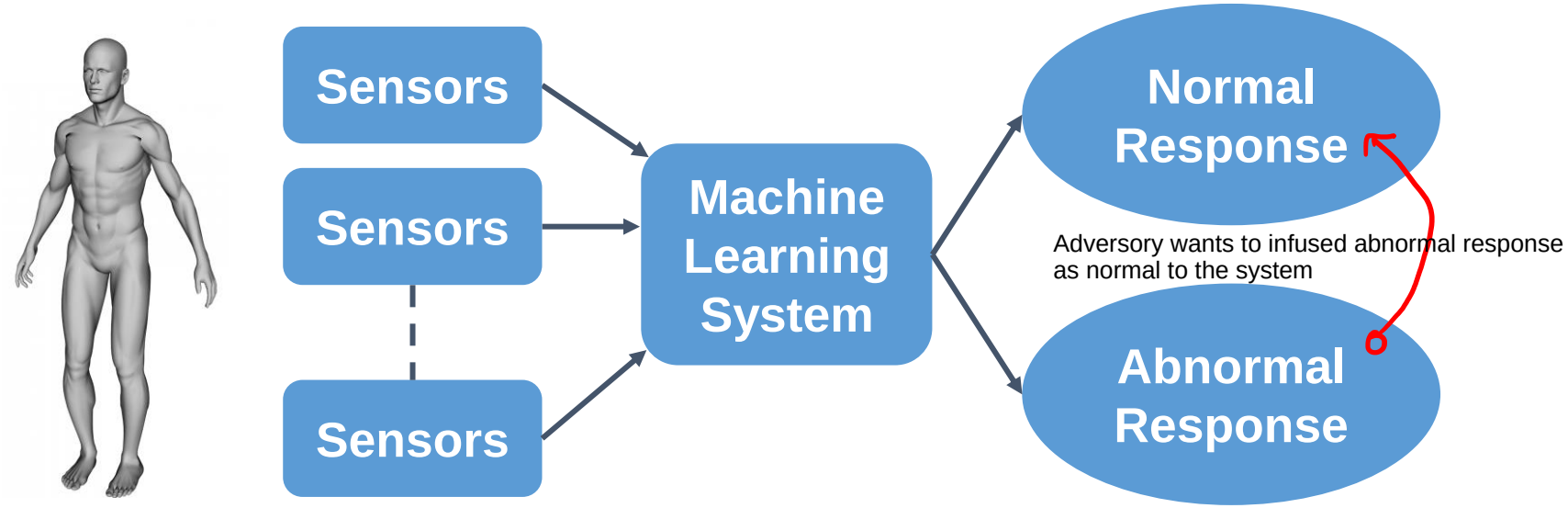


Aim is to get red squares calssified as gray circles (user A)
If decision bounday eventually changes, adversary becomes successful

Evasion Attacks

Cardiac & BP monitoring

Data alteration detectors



- Guess a feature that falls in the normal class
- Or alter a feature so that it falls in abnormal class

Adversarial ML Security Performance Tradeoff

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

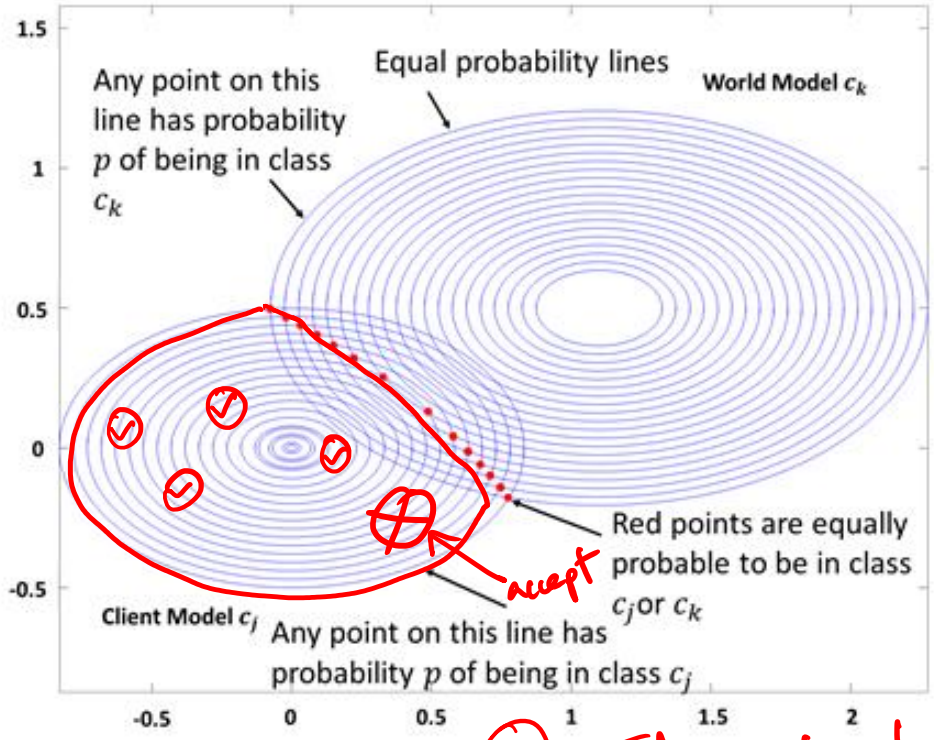
Evaluation of Machine Learning

ML performace vs security strength. They ahve tradeoff. Increase one, we lose on other

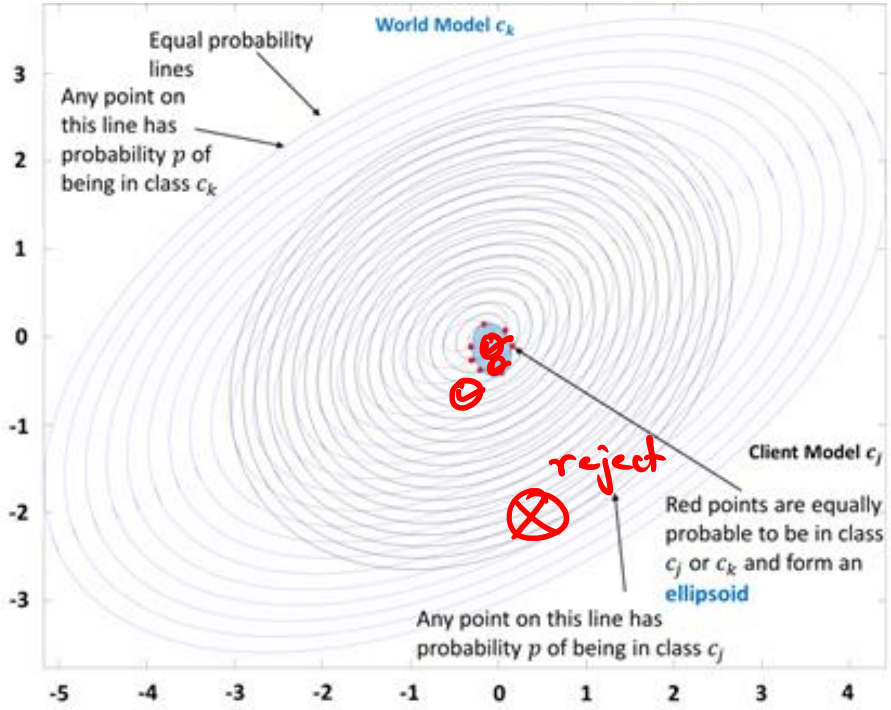
- **Security strength**
  - Many measures are possible
  - Effort required by adversary to guess a feature in a given class
- **Machine Learning Performance**
- **Tradeoff**
  - Security strength and performance

Tradeoff Example

Example for Naïve Bayesian Classifier



1 False Negatives ↓
Lower security



2 False Negatives ↑
Higher security

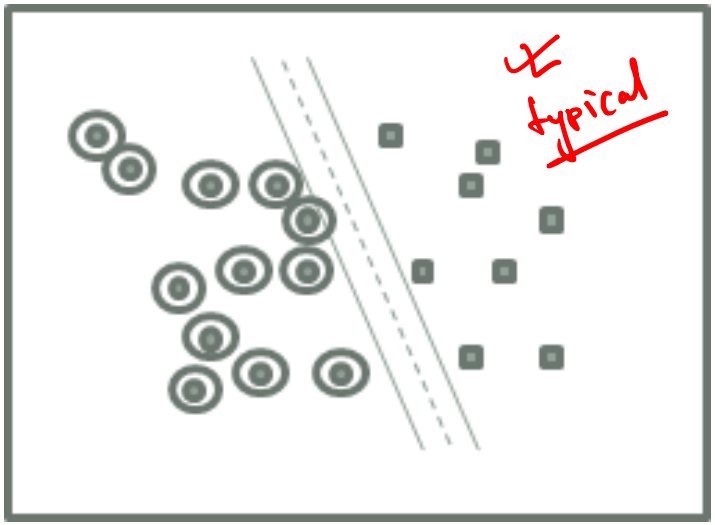
Performace of 2nd one is poor wrt first one but 2nd is more secure.

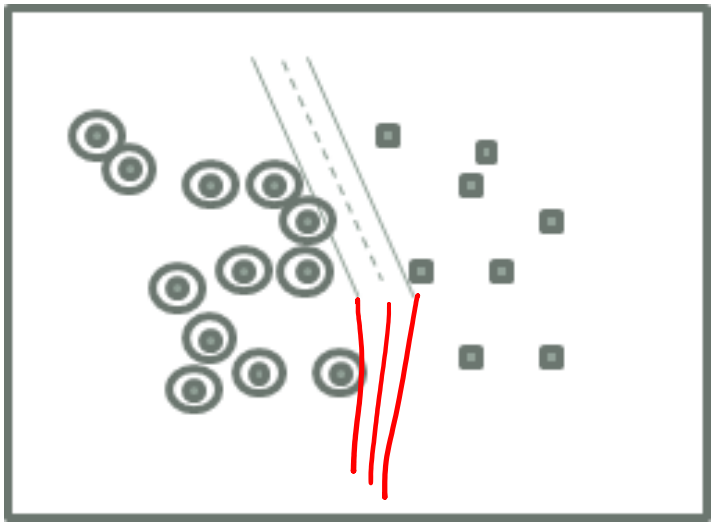
Hardening Technique

change machine such that its more robust to attacks

Measures to improve security of ML algorithms:

- Fitness check
- Increase complexity of classifies
  - Convex polytope SVM (decision boundary is not simple line, but it costs performance)





Security Performance Tradeoff

- Increase in security strength → hardening
- Hardening implies more difficult classification boundaries
- May increase **false** positives or negatives
- How to find a balance between security strength and performance?
  - Multi-objective optimization problem → difficult

Location Management

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Agenda

Location management

- Managing the mobile phone connectivity to the cell tower during user mobility

Earlier connectivity was majorly wired.

Fundamental Ideas

Any mobile phones are called mobile host

- **Mobile hosts (MH) are served by base stations (BS) [also called access points (AP)]** (mobile towers or wifi router)
- **Mobile hosts can roam about the network**
- **Mobile hosts (or other parties) must locate other mobile hosts to communicate with them**
  - Involves finding the base station currently serving the mobile host
  - **Search operation** Searching base station serving other device

Fundamental Ideas

- **When a mobile host moves, it must let the system know where it is**

  These two are separate actually: Registration done once when device come up, then periodically update operation takes place

  - Update operation (also called **registration**)
- **Must allow mobile hosts to switch between base stations to support roaming**
  - **Handoff operation**
- **MH will be served by one BS at a time**
- **BS coverage is one cell**

  (these are the assumptions we are making to keep things simple)

Location Management: Tradeoffs

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Tradeoffs: Location Information

There are several parameters that can be tweaked. We need to understand tradeoffs between several parameters. Changing one parameter might impact other parameter.

Location information is needed for search operation. Say A trying to connect to B

Location information can be maintained at various granularities:

If we know which cell is B connected to - We can directly connect. Easy search.

- **One cell** – requires MH to update location every time it moves from one cell to another
  - Tradeoff: More accurate location info vs. a large number of updates, which may overwhelm the system
- **Cell group** – organize cells into groups, only update when leaving current group
  - Tradeoff: Less accurate location info, which will require paging every BS in the group, fewer updates, and less load on the system

If B is highly mobile, to keep granularity to one cell, it has to make lots of handoffs which consumer too much power
So we can instead sacrifice location precision and instead deal with cell groups. Search is bit difficult now but power saving
Cell group might have say 10 cell towers. For search now, request comes to cell group and then that cell group searches for B in whole group. But now till the time B remains in this cell groups, it doesn't need to waste power in handoff
If B goes to another cell group, it will need to do handoff.

Tradeoffs

- **There will be always be tradeoffs between cost of search and update operations** (example power consumption)
  - More updates = more accurate location info = less cost for search
  - Fewer updates = less accurate location info = more cost for search

Location Management: Handoffs

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Handoffs

- **Handoffs between BSs are required to support roaming**
- **There isn't necessarily a one-to-one correspondence between handoffs and updates**
- **Issues:**

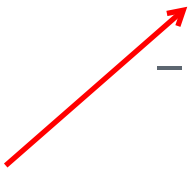
Hanoffs and updates are not same. Updates are not necessarily supporting mobility.
Even static phone runs update operation periodically.

  - When to handoff?
  - Selecting a new BS

If you send infor to cell tower but don't hear back, means you lost connection
But we can't wait till we lose connection. In between you might lose info.
- Another option is context aware handoff - phone monitors signal strength - if vicinity has better signal strength - do handoff irrespective of distance of cell tower. But here we are not using mobility context - what if you are moving towards your current cell tower. If you do handoff now, you will again need to connect back to current cell tower.

There might be many options, which one to connect to

  - Allocation of resources such as channels
  - Informing the old BS so that packets destined for MH can be forwarded



this can lead to overlaoding
of new tower T2

If we use location/distance from cell towers, we can do handoffs - but this again has problems
Say another cell tower T2 has better tech and lots of Mobile hosts might connect to T2 making it overloaded.
So signal strength, location (distance), mobility and load on tower are some of the involved factors in Handoff decision. Dynamic evnvironment makes decison making complex.

Mobile operator companies incur cost in search, registration, update, handoff and maintainence - that's why they charge users.

Handoffs

Request generated by mobile

request generated by network

- **Mobile controlled handoff vs. network controlled handoff**

  Which one of above is used depends on the parameter sued for making handoff decision

- **Handoff may be necessary because:**
  - Mobile host is moving — Mobile controlled handoff
  - Current BS is overloaded — Network controlled handoff
  - Quality of communication with current BS is poor

    Mobile controlled handoff because phone only know QoS

Handoffs

Choosing a new BS:

- Based on signal strength
- Base on predicted movement of MH
- Based on resources available at BS

So its a multi-variable tradeoff based multi-objective problem.

Location Management: Location Registrars

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

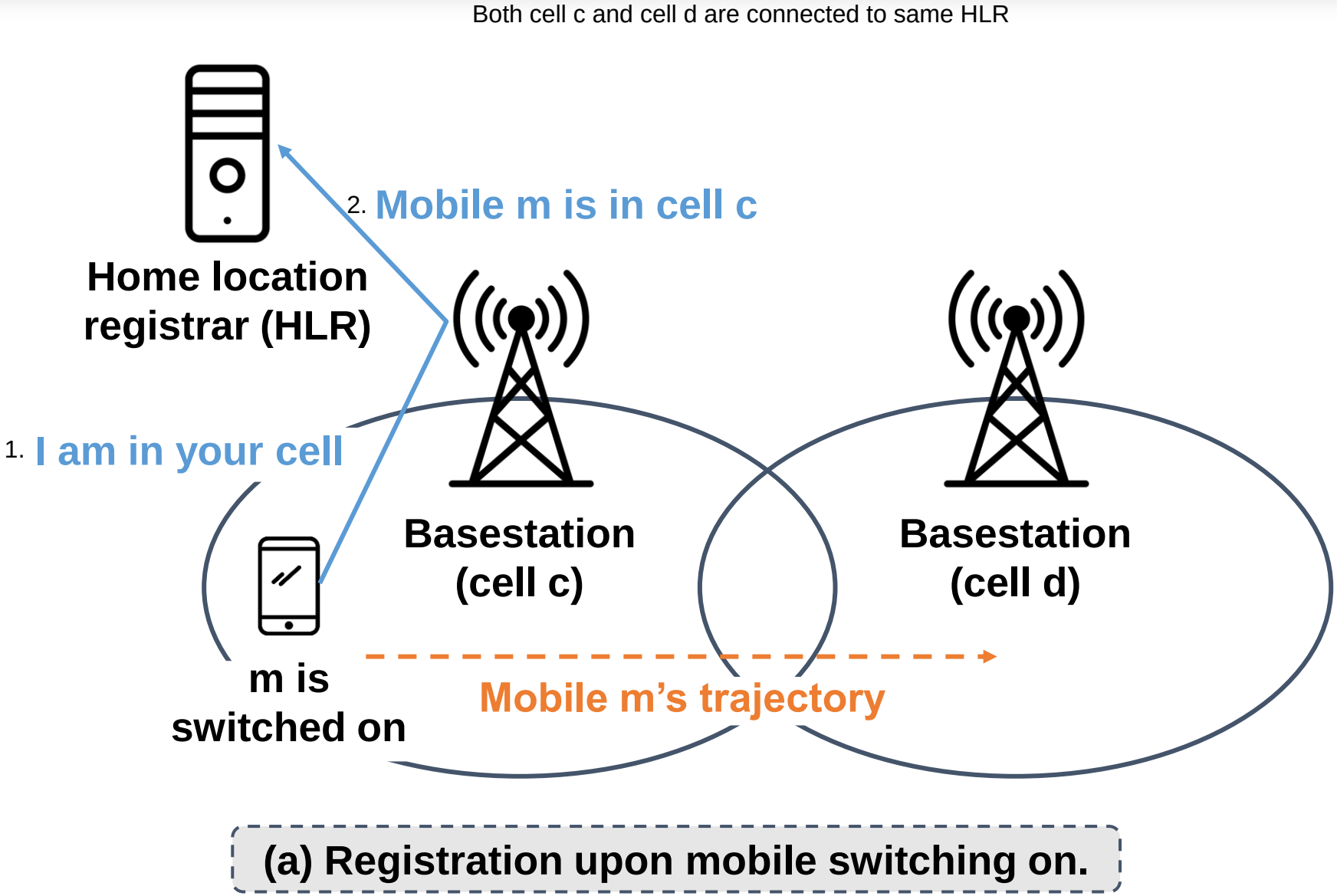
Location Registrars

- **Location Registrars (LR) are databases containing location information for MHs**
  - Can be one or many throughout the network
- **To get the idea, consider a system with only one LR, a Home Location Registrar**

  This is the LR which MH first connected to when it came up and registered.
- **Location is maintained at single-cell granularity**

Single HLR can handle multiple cell towers. So cell towers directly connect to MH but HLR are one level above them.

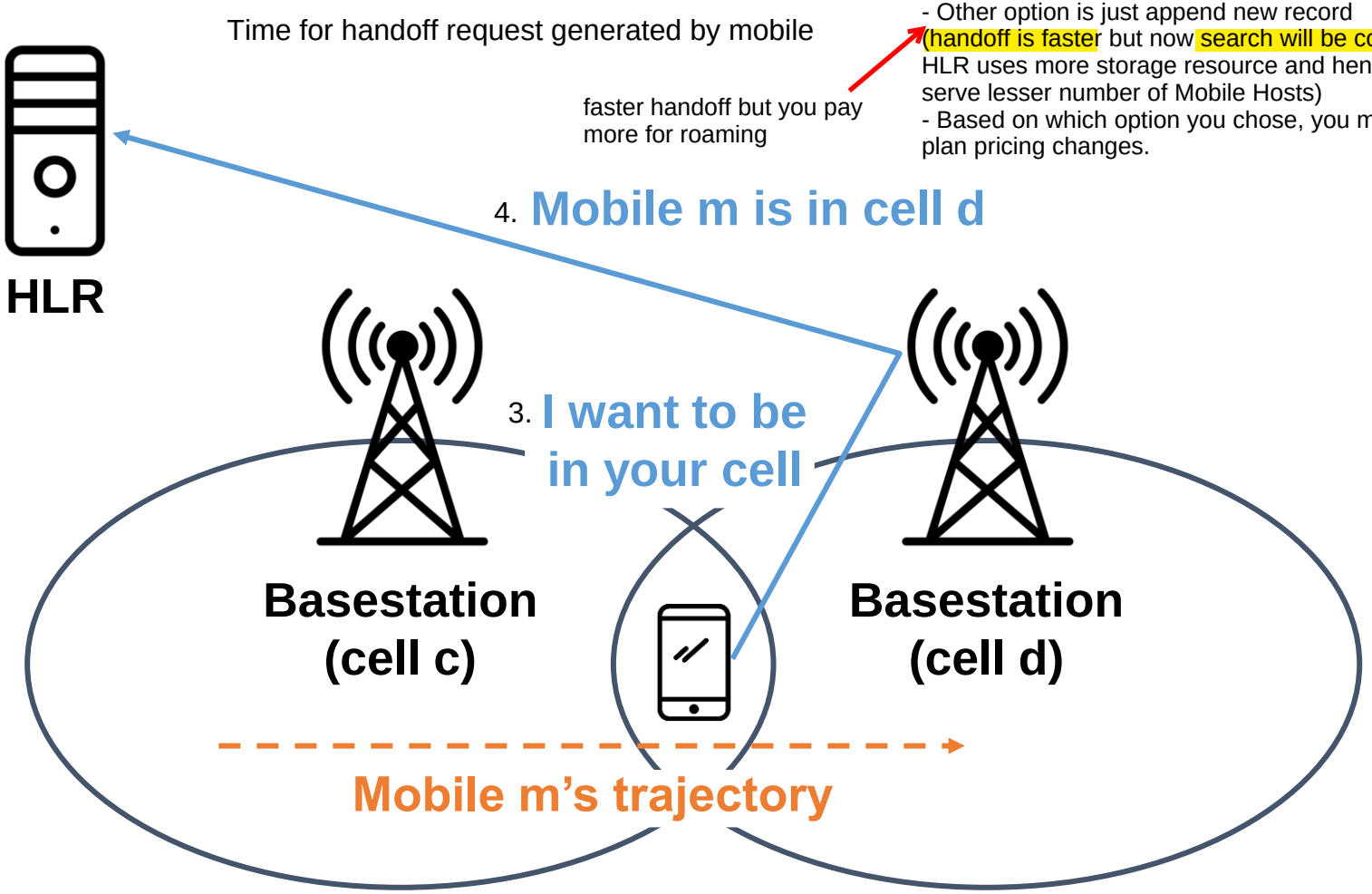
Single LR Scheme: Switching ON



(a) Registration upon mobile switching on.

Single LR Scheme: Handoff

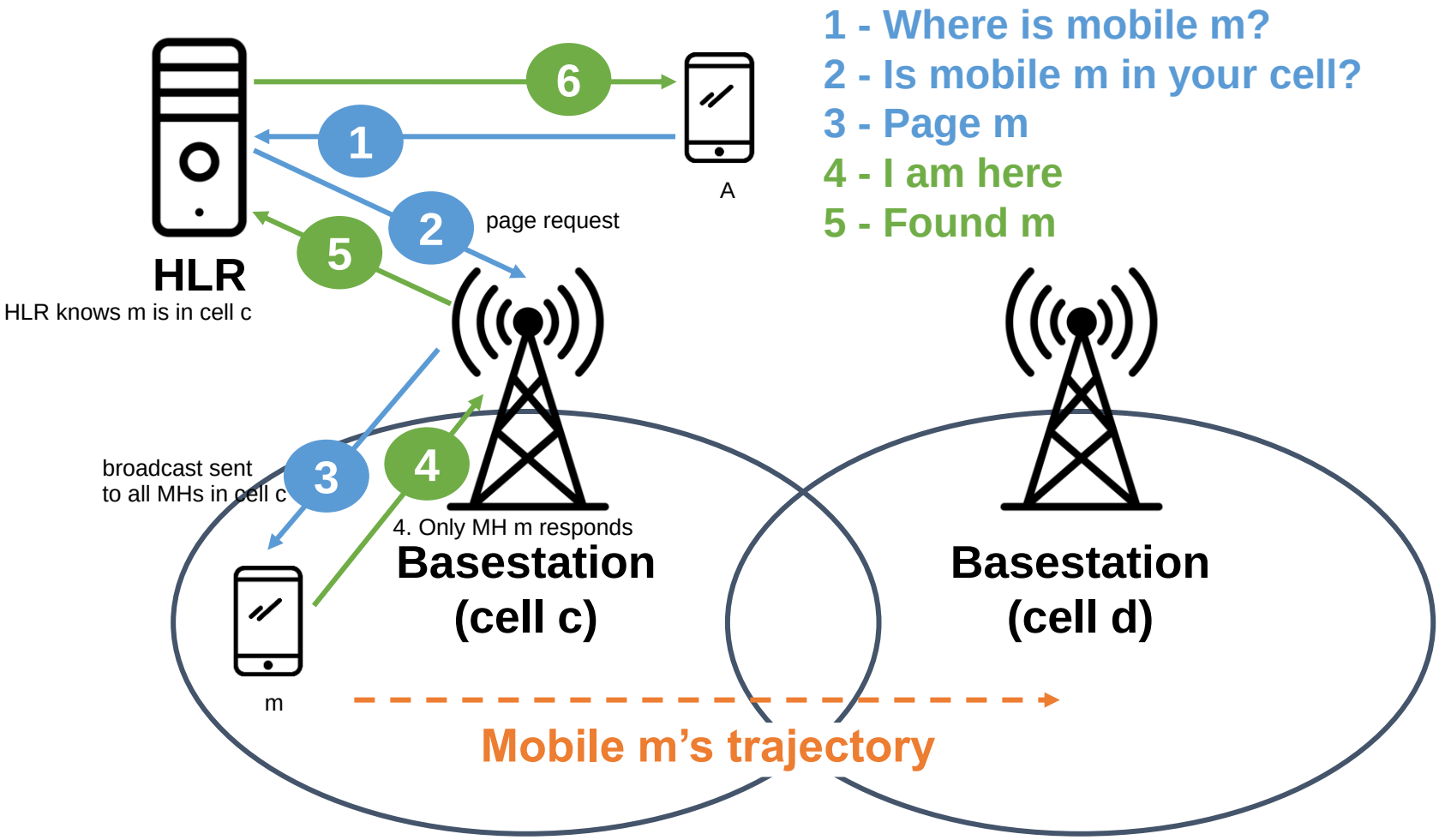
- One option HLR has is to overwrite old entry (for this HLR needs to search its whole DB - thus handoff takes more time)
- Other option is just append new record (handoff is faster but now search will be costly + HLR uses more storage resource and hence can serve lesser number of Mobile Hosts)
- Based on which option you chose, you mobile plan pricing changes.



(b) Registration upon cell handoff.

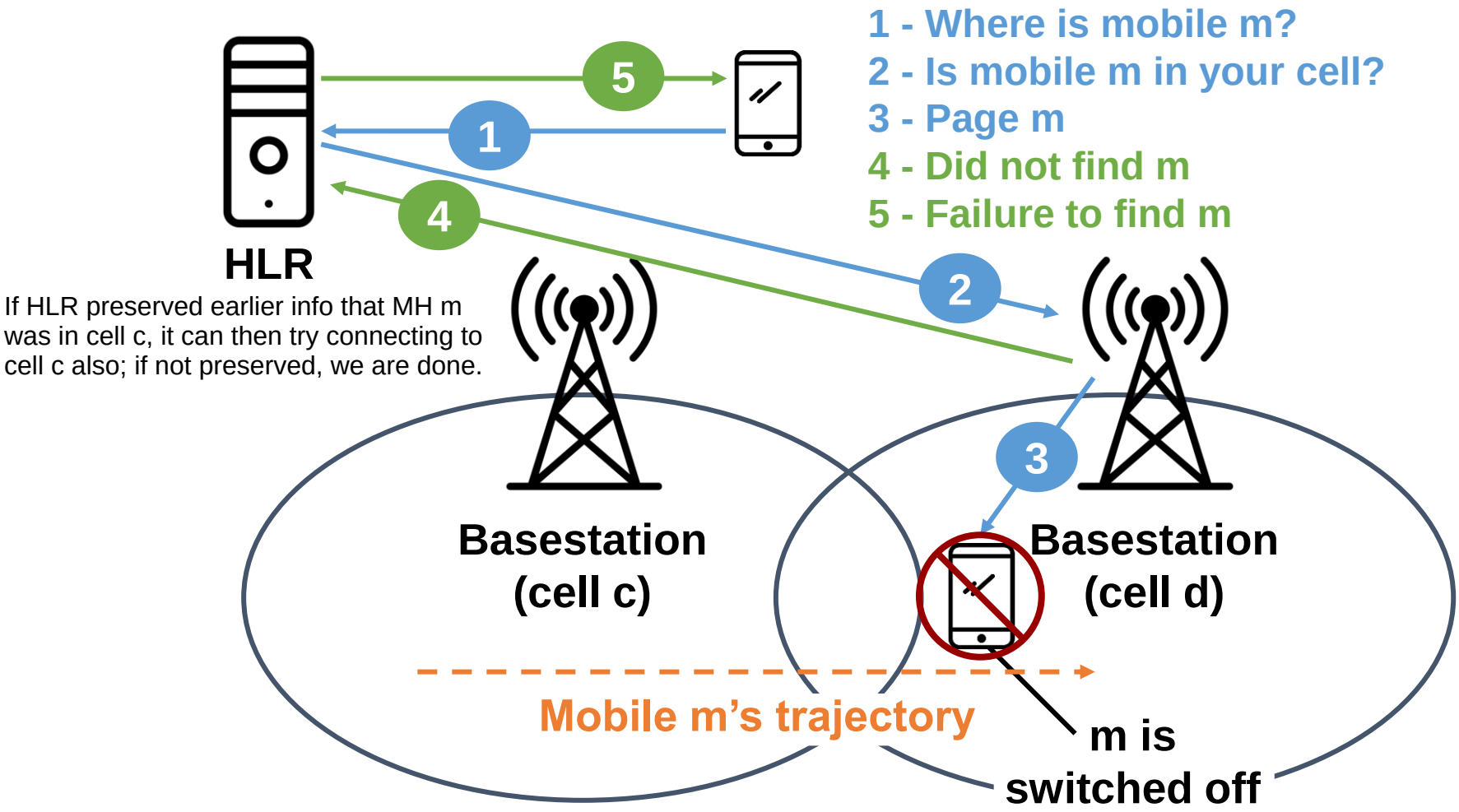
Single LR Scheme: Search success

Host A searching for Host m



(c) Another mobile wants to find m - success case.

Single LR Scheme: Search Failure



(d) Another mobile wants to find m - failure case.

Location Management: Enhancements

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Enhancements to Single LR Scheme

TTL is a timespan and cell tower would expect an update from MH every TTL. If not received - either MH is dead or not connected.

Can add a timestamp and time-to-live (TTL) to registration information

- TTL (not the same as residence time of node in cell)
  - Helps in constraining the search cost
    - Search diameters = max speed x TTL

if MH is not responding, we can search neighboring cells but how many we should search?

So we constrain search diameter

max speed of MHs in that area
Cell tower haven't got any update in this TTL. How far could the MH go in this TTL? Search that area.

- Related to concept of soft-state
  - Hard-state – explicit revocation of “state”

overwrite old location in HLR

  - Soft-state – implicit revocation of “state”

preserve old info also in HLR

    - Makes the system more fault-tolerant – adaptive to changes in system (but this needs more resources)

Enhancements to Single LR Scheme

- **If time-to-live expires, then location for mobile host is assumed out of date** (we can then free up memory used by this MH records)
- **Can expand the search to neighboring cells when attempting to locate a MH**

Location Management: Registration Area

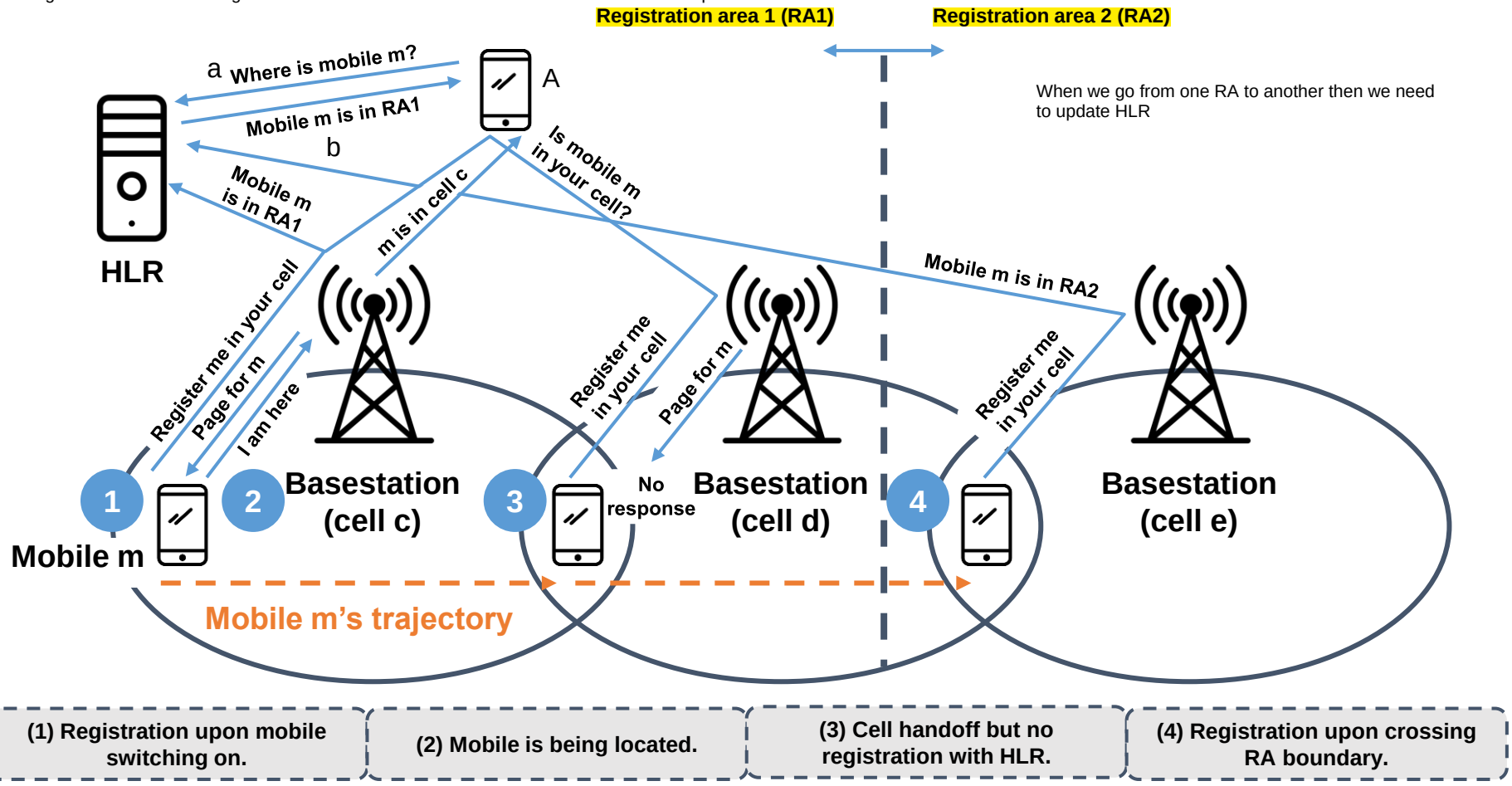
Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Registration Area-Based Location Management

It partisions cell towers into different groups

Step c: A asks all cells in that registration area if they have MH m. All cells in that registration area page all their MHs to see if m is here. M responds and then A and m can directly connect
Advantage here is in such setting - if MH m moves from cell c to cell d => we don't need to update the HLR



Registration area = A group of cells cells means cell towers

(update only when crossing a registration area boundary)

Properties of RA-Based Approach

Advantages

- Reduces update cost (no need to update HLR if moving within same RA)
- Bounds the search cost (number of cells queried)
  - Search is restricted to an RA
- Makes LM more scalable
  - By reducing number of registrations to be processed by the HLR
- Makes LM more manageable (as compared to per-user LM schemes)

Properties of RA-Based Approach

Issues:

- Granularity of RA (how to configure system into multiple RAs)

Which cells do we group together?

- Same size (homogeneous) or different size (in terms cells)?
- Optimization problem: How to partition the cells into RAs taking into account call + mobility pattern so as to optimize “LM cost”

Answer depends on which type of lcoation we are looking at.
- Say some area has high density and industry - there will be lots of phone calls - we can have more number of cell there
A small physical area can have lots of cells and be a single Registration Area
- For sparse populated residential area, we can have lesser cells and RA can be bigger.
This is pattern-based partitioning.

We can partition based on mobility pattern also - say m keeps moving between cells c and d, then it makes more sense to keep cells c and d in a single Registration Area - else HLR will face too many handoffs. This is personalized RA.

In search setting search call is more but someone finding you doesn't cost you much (some companies have expensive outgoing)

Other Optimizations

- **Movement-based update (non-RA based)**
  - Update location when MH crosses a specified number of cell boundaries
- **Distance-based update (non-RA based)**
  - Update location when MH moves a specified distance away from the last point of update
- **Time/Movement/Distance-based schemes are per-mobile user based schemes.**
- **In contrast to these, RA based scheme looks at aggregate mobility and call patterns**

Location Management: Multiple Registrars

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Multiple Location Registrars

(TIll now we dealt with single HLRs)

Forwarding pointers

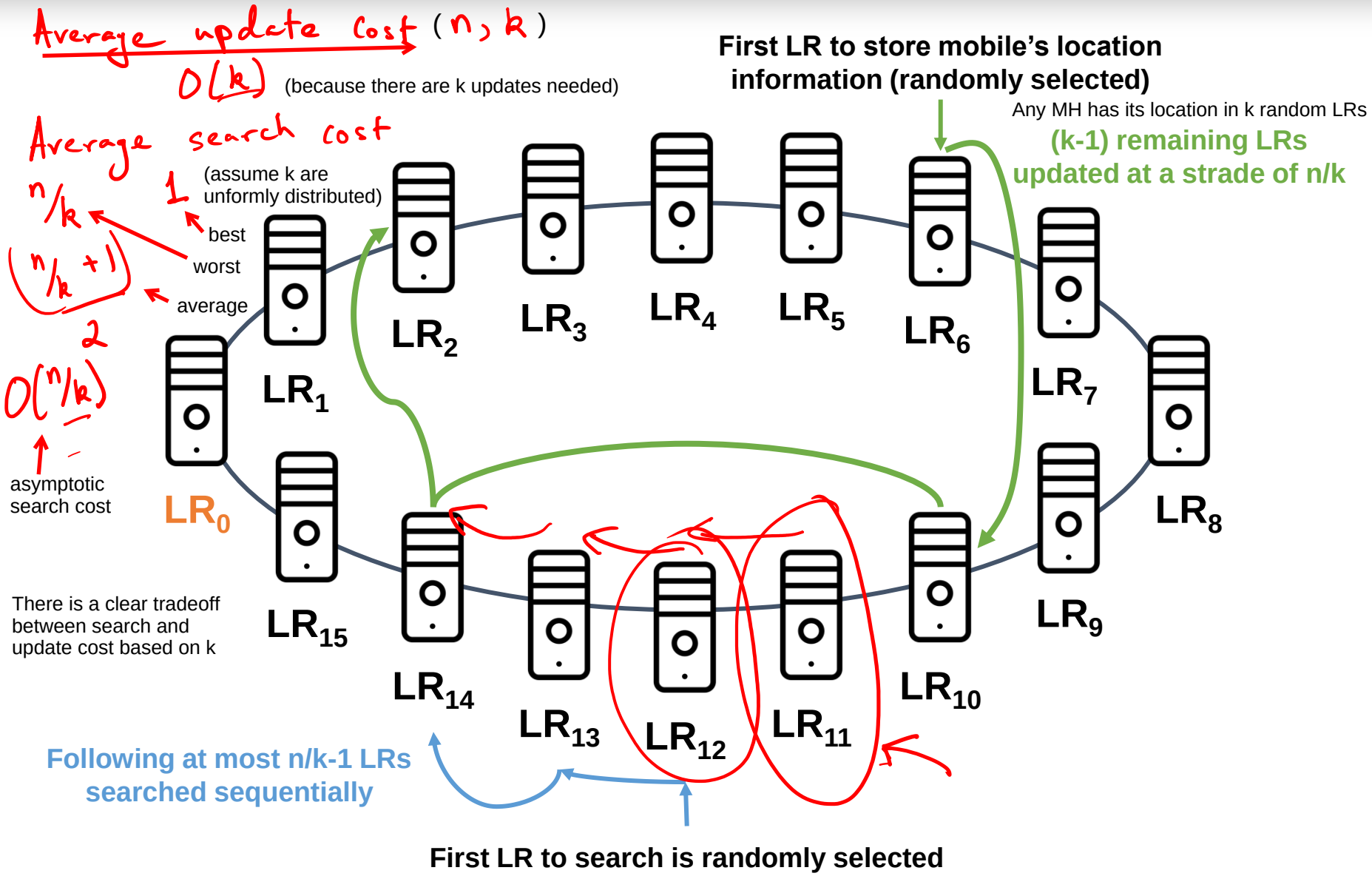
- When maintaining multiple location registrars, use a chain of forward pointers to track the MH like linked list pointer

Replication of location registrars

- Flat
- Hierarchical

Flat Replication

All LRs here are at same level



Hierarchical Replication

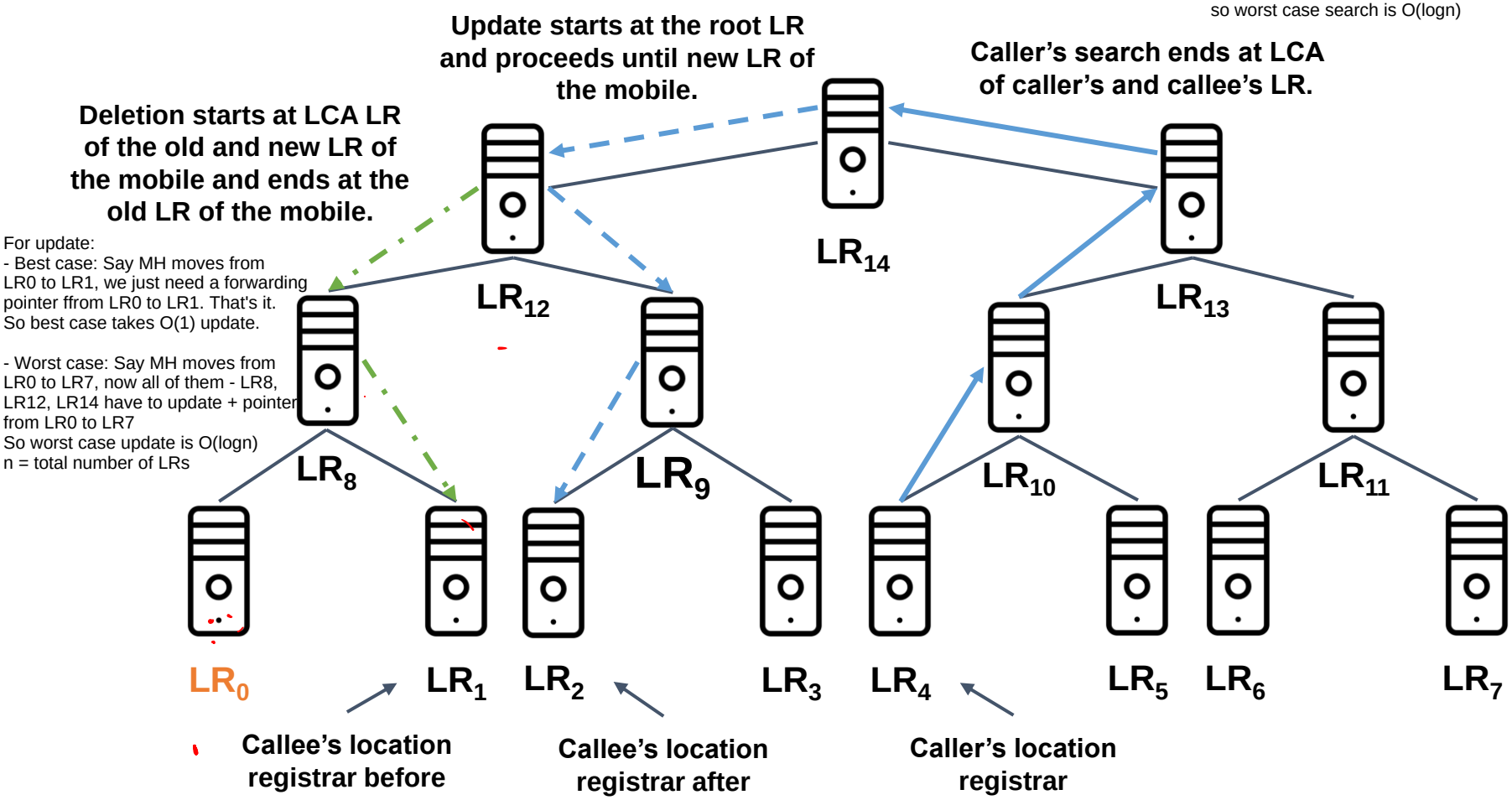
arranged like tree
Leaf nodes directly connect to cell towers

Here both search and update take O(lgn) in worst case and constant in best case

LR14 will need very huge resources

Non-leaf nodes cache all info in attached subtrees

For Search: m searching B
- Best case: Say m is in LR0 and B is in LR1, We got to LR8 and from there jump to LR1 So best case search takes O(1)
- Worst case: Say m is in LR0 and B is in LR7, we go to LR8->LR12->LR14->LR7 so worst case search is O(logn)



A hybrid of flat and hierarchial may be used.
Eg: Connect all such network roots in flat manner

Mobile Internet Protocol (IP)

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Earlier there were ethernet cables only. But now we have wireless.

We connected to LANs using ethernet. Wifi eleminated need of LAN and replaced it with WLAN.
Range of wifi is limited but we want to create unlimited reachibility.
Instead of using router as Access point, why not connect to cell tower as Access Point?

Ira A. Fulton Schools of Engineering
Arizona State University

Why Mobile IP?

- A protocol is needed which allows network connectivity across host movement
- Protocol to enable mobility must not require massive changes to router software, etc.
- Must be compatible with large installed base of IPv4 networks / hosts

DNS server does name -> IP conversion.
That was in IPV4 but it doesn't work when there is lot of mobility. In wifi at least your modem had fixed IP.

While working with wifi, change in IP address is not so frequent as it is with mobile.

Each time your Mobile host updates its IP, DNS must be updated so that other devices on internet can reach you back.
In mobile roaming you often update IP address so frequent DNS updates wil happen.
Imagine so many mobile soing so many updates => DNS can't handle it.
So we handle mobile internet with IPV4 technology. We need to move away from it.

Why Mobile IP?

- **Confine changes to mobile hosts and a few support hosts which enable mobility**
- **Just hacking DNS won't work:**
  - DNS updates take time
  - Hooks for normal users to update DNS won't be accepted by administrators
  - After DNS lookup, raw IP address is used by TCP, UDP, …

Internet Protocol (IP) (IPv4)

(IPv4 TCP works based on ACKs. I we don't update DNS records, other host will keep sending oackets to old IP address because they are not getting ACK)

- **Network layer, "best-effort" packet delivery** (works based on ACK)
- **Supports UDP and TCP (transport layer protocols)**
- **IP host addresses consist of two parts:**
  - Network ID + host ID
- **By design, an IP host address is tied to a home network address**
- **Hosts are assumed to be wired and immobile**
- **Intermediate routers look only at a network address**
- **Mobility without a change in IP address results in unrouteable packets** (and we can't support so frequent updates)

Mobile Internet Protocol (IP) Basics

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Mobile IP: Basics

allows mobile MH to mmove through network seemlessly
No large DNS updates required
No unnecessary ACK waiting

- **Proposed by IETF (Internet Engineering Task Force)**
  - Standards development body for the Internet
- **Mobile IP allows a mobile host to move about without changing its permanent IP address**

This permanent IP address could be that of a cell tower in home network

- **Each mobile host has a home agent on its home network**

IP addrss of home address can be permanent IP addrss of Mobile host
Permanet IP addrss can change but not so often.

- **Foreign agents on remote networks also assist**

Foreign agents also assits in data transfer

- **Mobile host establishes a care-of address when it's away from home**

When mobile host is away from home addrss, foreign host (may be a cell tower) give a care-of address.

Mobile IP: Basics

say entity B trying to reach Mobile Host m

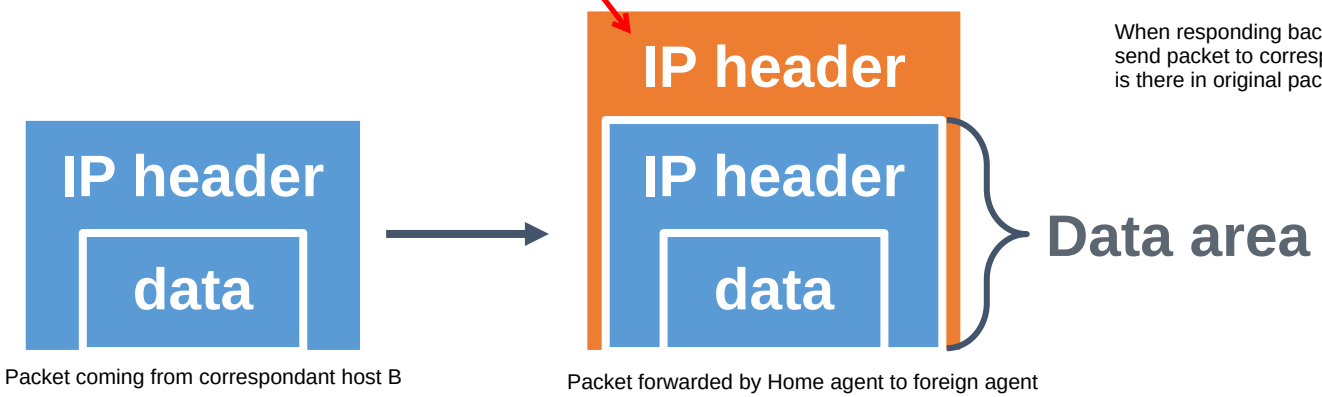
- **Correspondent host** is a host that wants to send packets to the mobile host
- Correspondent host sends packets to the mobile host's IP permanent address
- These packets are routed to the mobile host's home network
- **Home agent forwards IP packets for mobile host to current care-of address**

  We use forwarding pointers to let home agent know that MH has moved and now has a new care-of address. Home agent use this pointer to forward packets to MH's current care-of address
- Mobile host sends packets directly to correspondent, using the permanent home IP as the source IP

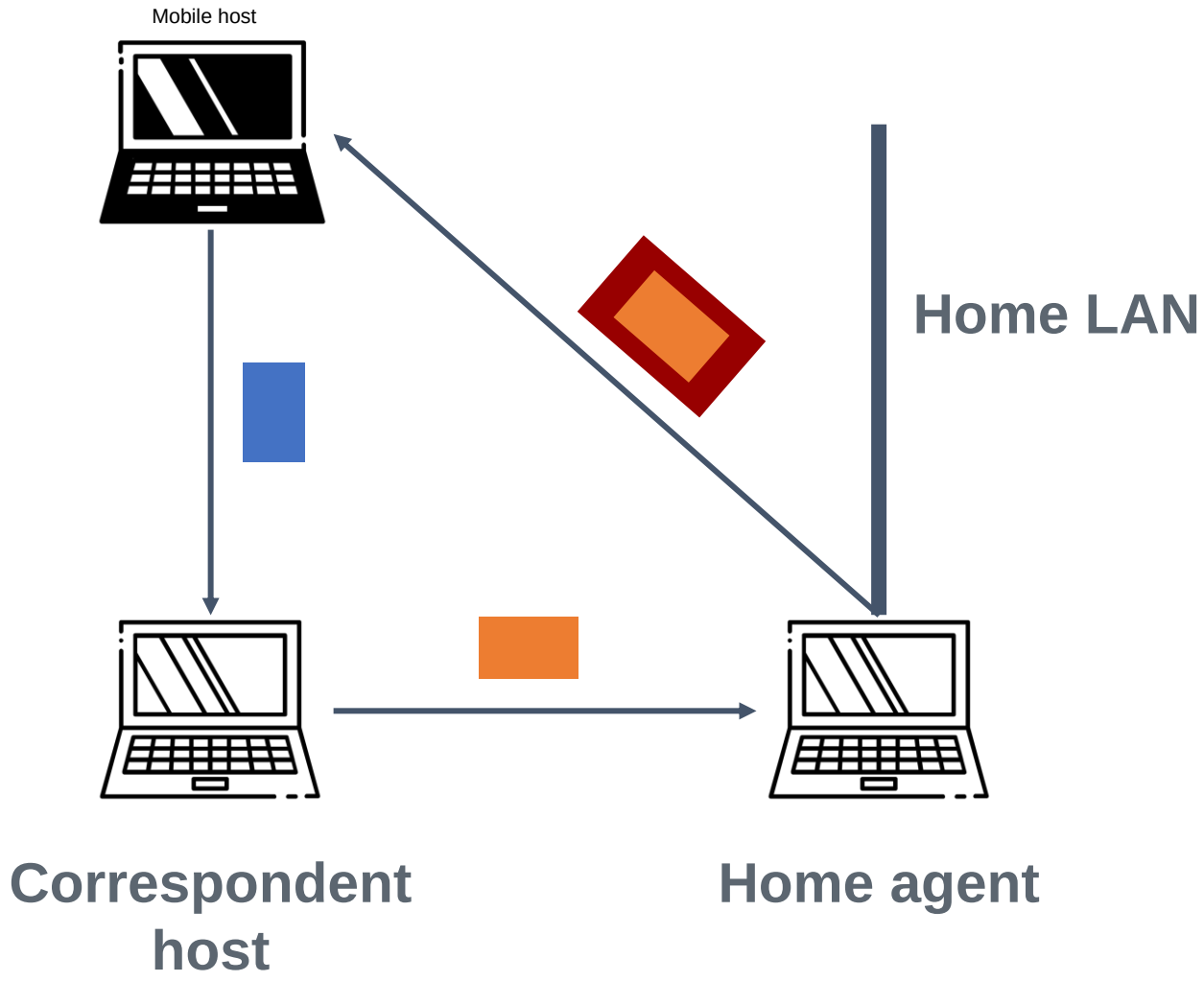
When Mobile Host moves to foreign host (doesn't matter how frequently), it updates Home agent using forwarding pointer. So we don't need any DNS updates now.

Aside: IP-in-IP Tunneling

- Packet to be forwarded is encapsulated in a new IP packet
- See RFC 2003 for details
- In the new header:
  - Destination = Care-of address
  - Source = Address of home agent
  - Protocol number = IP-in-IP



Mobile IP: Basics



Mobile IP Nuances

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Protocol Messages

In addition to data, we need some protocol control messages as well. We need extension for ICMP for example

A home agent should advertise. Foreign agent should also advertise - connect to me. These are example control messages.

- **Extensions to Internet Control Message Protocol (ICMP)**
- **Agent advertisement**
  - "I'm a foreign agent"
  - "I'm a home agent"
  - Agent advertisements seen by mobile hosts on their home network welcome them back
- **Agent solicitation** apart from advertisement, we have agent solicitation
  - Mobile host actively seeks foreign agent
  - Elicits agent advertisement message

Protocol Messages

- **Registration Request** When mobile host comes up first time
  - Sent to home agent
  - New IP address
  - Flags to indicate whether broadcast traffic should be delivered
  - Security information to prevent remote redirects/replay attacks (more soon)
- **Registration Reply**
  - ACK or an error

All these need control messages
in additional to actual data packets
They add delay and overhead

Care-Of Addresses

How MH gets care-of address

- **Whenever a mobile host connects to a remote network, there are two choices:**
  - Care-of can be the address of a foreign agent on the remote network
    - Foreign agent delivers packets forwarded from home agent to mobile host
  - Care-of can be a temporary, foreign IP address obtained through, for example, DHCP
    - Home agent tunnels packets directly to the temporary IP address

foreign agent get temporary address from DHCP for that MH
These temporary addresses can be reused for other hosts later

- **Regardless, care-of address must be registered with home agent**

At the Other End

In IP-in-IP tunneling someone has to uncover original packet for Mobile Host. It depends on type of care-of address used:

- **Depends on the type of care-of address:**
  - Foreign agent's IP, or (if we used foreign agent address as care-of, its foreign agent task to uncover - adds load)
  - Mobile host's IP (**Remote IP** obtained via DHCP) here mobile host directly handles decapsulation
- **Someone strips outer IP header of tunneled packet, which is then fed to the mobile host**
- **Decapsulation can be performed by agent or mobile host**

So its a trade-off here as well.

At the Other End

Aside:

- Any thoughts on advantages of foreign agent vs. co-located (using foreign agent's IP) address?
- Which has less overhead for mobile host?
- Which consumes fewer IP addresses (still a concern with IPv4)?

We are getting new and new protocols becasue we are changing these involved nuances.
Multiple factors and tradeoofs calls for newer and newer protocols.

Mobile IP Inefficiencies

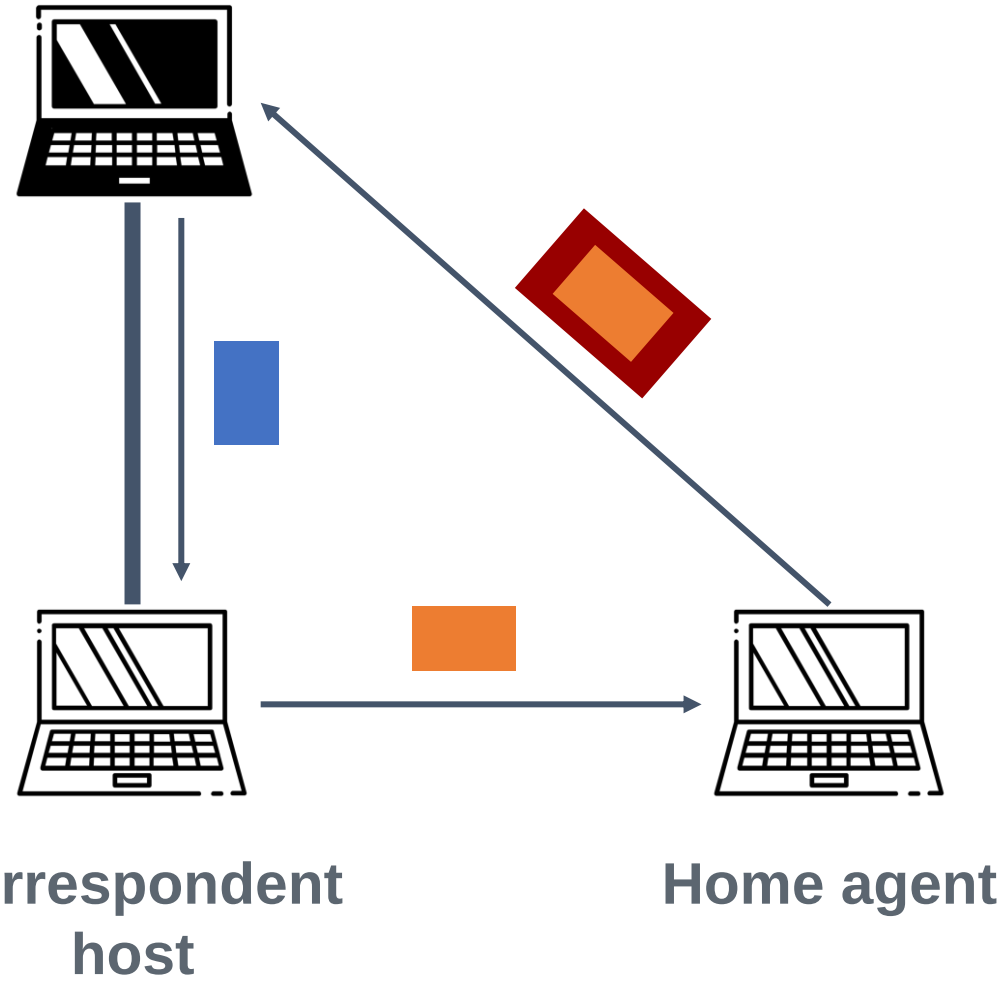
Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Routing Inefficiency

What if correspondant host and mobile hsot are in same home agent? - they still go via home agent because they are not aware of being in same home agent area.

Mobile host and correspondent host might even be on the same network.



Route Optimizations

Possible solutions:

- Home agent sends current care-of address to correspondent host

  Once home agent is aware of topology, it can ask participating agents to communicate directly

- Correspondent host caches care-of address
- Future packets tunneled directly to care-of address
- Problems when mobile host moves…

  When MH comes back from foreign agent to home agent network, when does it update that forwarding pointer and all that cached info with correspondant etc.

- Care of address becomes stale, needs to be updated

  Too frequent updates consume too mcuh power - again a tradeoff involved

- Requires that correspondent hosts understand Mobile-IP
- Requires security relationship between correspondent hosts and home agent of roaming mobile host

Either home agent has to consume too much power because we don't want correspondant caching care-off address.
If they cache then correspondant waste power updating stale care-off addresses frequently. That's a tradeoff.

Mobile IP Remote IP

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Ira A. Fulton Schools of Engineering
Arizona State University

Remote IP

(When we get IP from foreign agent from pool of temporary IPs)

How does the mobile host get a remote IP?

- Router advertisements, DHCP, manual
  (foreign agent)
- Agent advertisement if remote network is mobile-IP enabled

In essence, foerign agent advertises, mobile host requests and they come together to fulfill this

How can a mobile host tell where it is?

How will MH know i need to request new remote IP?

- Am I at home? Am I visiting a foreign network? Have I moved?

One way: By listening for advertisements from its home agents

MH keeps sampling advertisements; if its from its home agent means it has not moved, else moved.
If I didn't hear from home agent in last 20 seconds, may be MH has moved

- Presence indicates home
- Absence tends to indicate not home…

Remote IP

What if Home agent dies because of any reason?

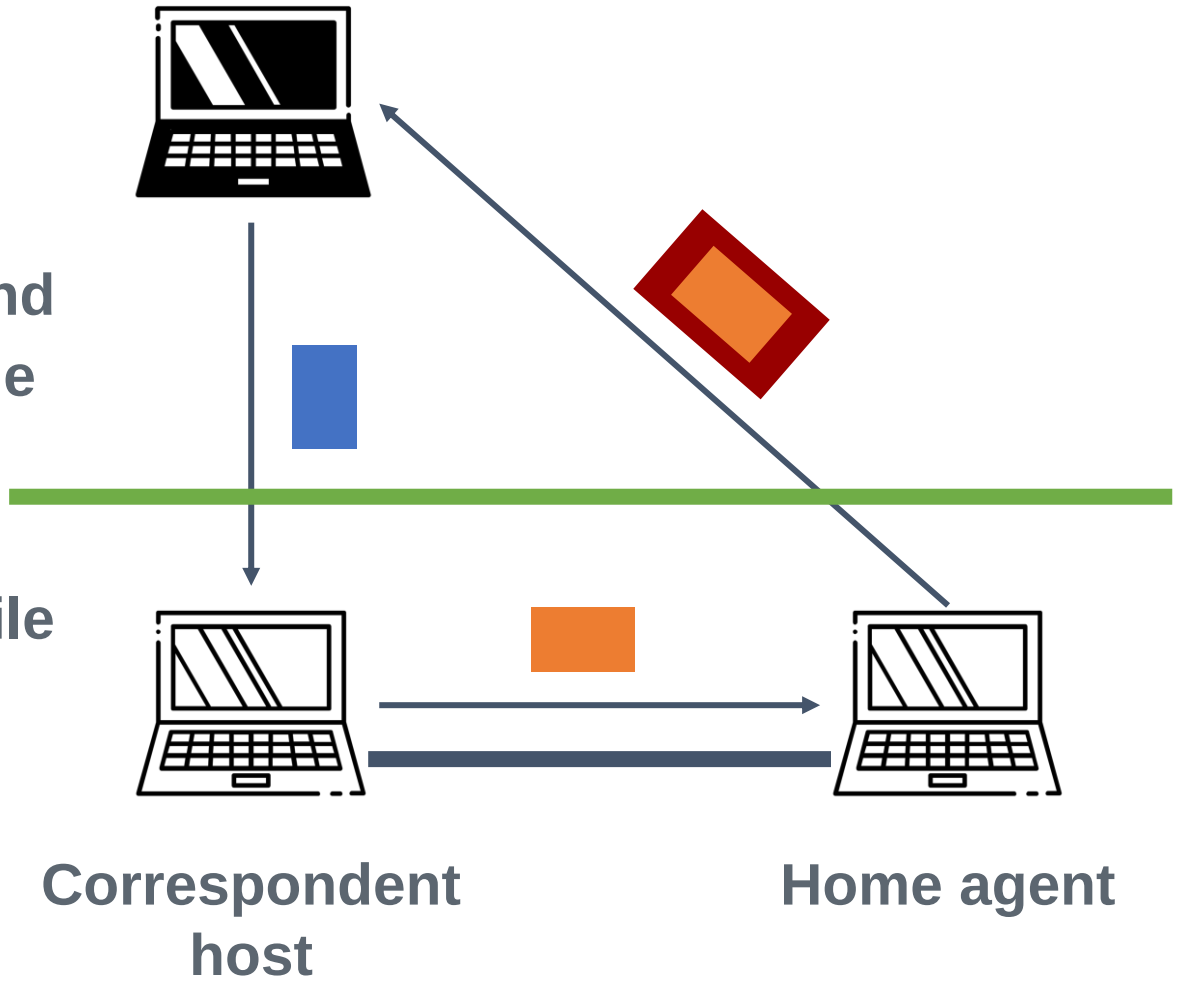
- **Redundancy: What if the home agent does not answer a registration request? Or it is dead?**
  - Registration request to broadcast address of home network
    made by Mobile host
  - All available home agents will hear and reply, but will reject service because message was received via broadcast
  - Error in Registration Reply (a rejection) carries new home agent ID
    produced by Mobile Host
  - Now can request help from a specific new home agent
    some foeighn agent can become new home agent
- **"Ingress" filtering** This problem occurs when you have a proxy which monitors which websites you are accessing
  - Routers that see packets coming from a direction from which they would not have routed the source address are dropped

Packets Dropped: "Ingress" Filtering

proxy server has list of topologicallly incorrect IP address, we ignore those
in Mobile IP, a foreign agent Ip address, or remote IP might be wrongly deemed as incorrect IP address and blocked.

Correspondent and home agent on the same network.

Packet from mobile host is deemed "topologically incorrect."



Mobile IP Security

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

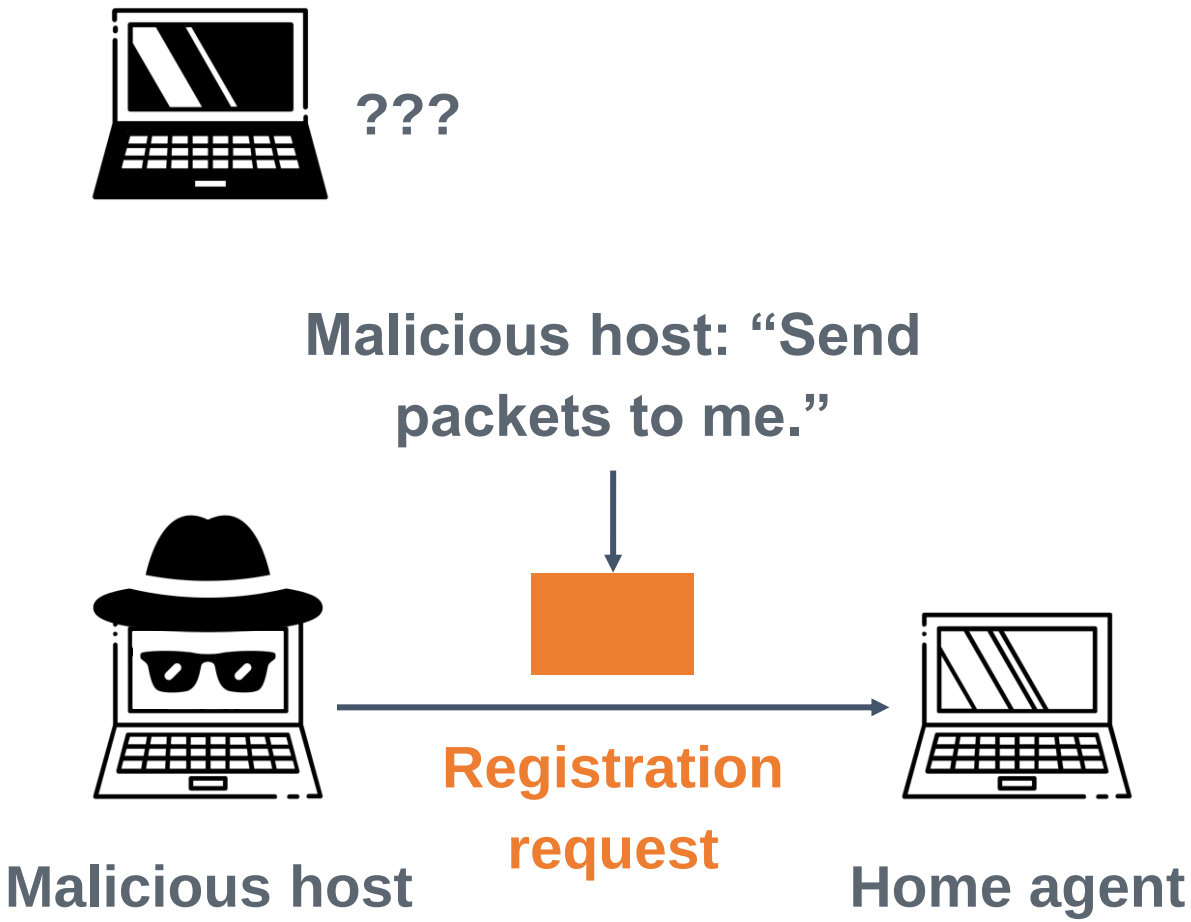
Ira A. Fulton Schools of Engineering
Arizona State University

Security Issues

- **We'll look at one of several security issues:**
  - Bogus registration (denial of service) attacks
    - Malicious host sends fake registration messages to home agent "on behalf" of the mobile host
    - Packets could be forwarded to malicious host or to the bit bucket

Malicious host wants to hijack packets intended for original mobile host

Bogus Registration Attack



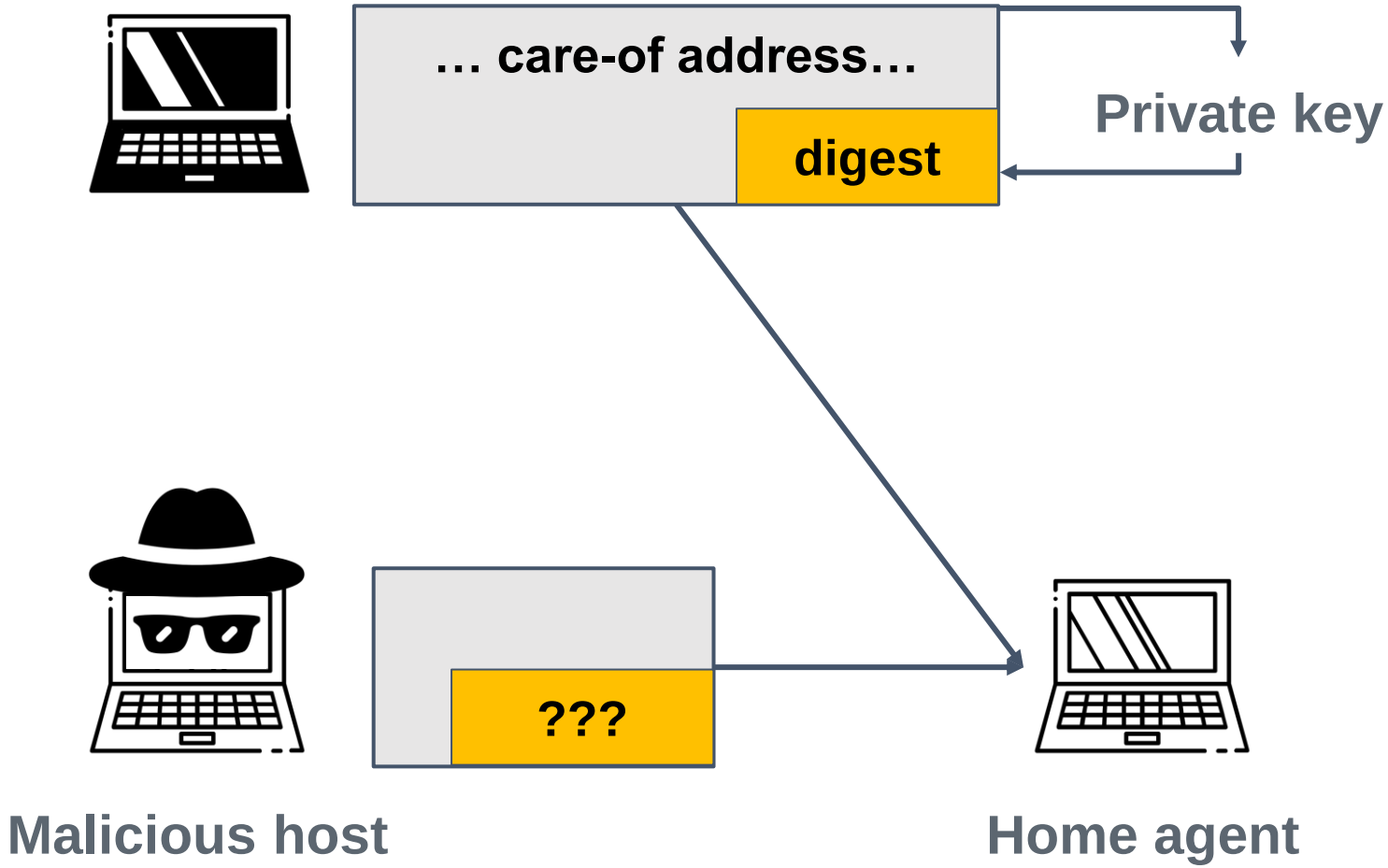
Authentication

- To fix this problem, authenticate registration attempts
- Use keyed message hashing to generate a message digest

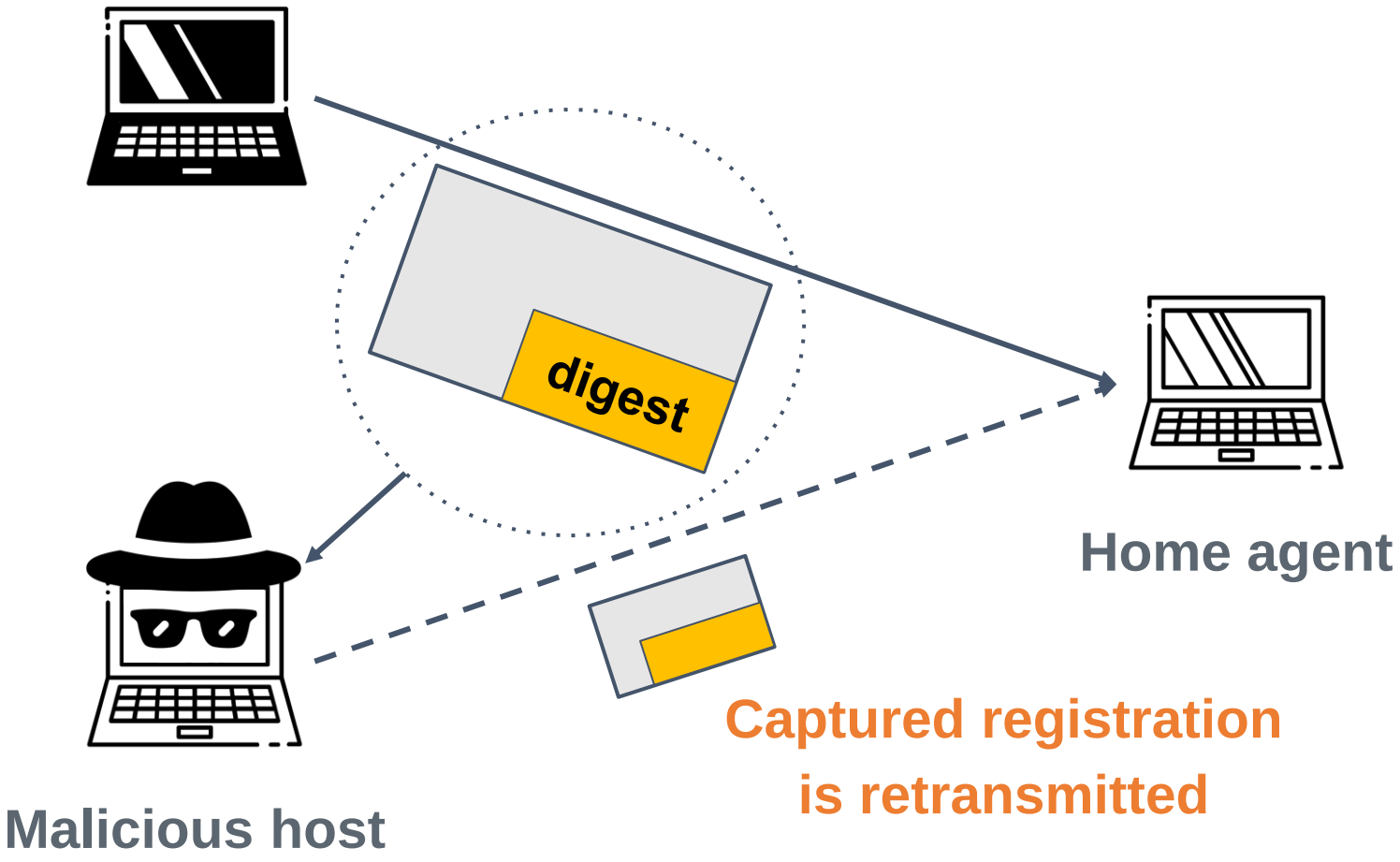
  Message should have unique id from mobile host which only home agent and mobile host knows;
  That unique id becomes encryption id. But such encryption has replay attack problem.

  - MD5: See RFC 1321
- Home agent generates hash using shared private key to message and see if the message digest is identical

Authentication



Replay Attacks



Avoiding Replay Attacks

- **Avoid replay attacks by making registration requests unique**
  - Add time or a pseudo-random number to registration request/reply
  - If time or random number is out of sync, provide info to resync in rejection
  - Insufficient information to help malicious host
- **Counter instead of time/random number is not as good of a strategy**
  - Would allow storing a ‘set’ of registration requests

Mobile IP Address Resolution Protocol

Ayan Banerjee, Ph.D.
Assistant Research Professor
Arizona State University

Web server trying to send info to host. It will send infor to router IP address.
After first communication, server can send packets directly to host if it know the MAC address of host
(now no need to go through router) - that's the purpose of address resolution protocol

But in mobile setting, mobile host may not be in home address now, how can home agent do address resolution?
So now home agent instead does proxy address resolution

Address Resolution Protocol (ARP)

- Allows hosts to broadcast an IP address and retrieve the MAC address of the host
- Home agent must perform **proxy ARP** for registered mobile hosts that are away

  it masks whole forwarding process
- Home agent must perform **gratuitous ARPs** when mobile host leaves home network to update ARP caches of local hosts

  When mobile host is in foreign network, home agent pushes the new MAC id of mobile agent to correspondant host even if it didn't ask for it. - gratuitous ARP
- Mobile agent, on returning home, must issue gratuitous ARPs for the same reason

  Gratituous ARP needed again when it comes back home so that correspondant host doesn't waste time hitting wrrong MAC id. It increases responsivenss of mobile IP.

Conclusions

- Great potential for mobile application deployment using mobile IP
- Minimizes impact on existing Internet infrastructure
- Security issues are important
- Firewall solutions proposed, but most are complicated
- Several working implementations (e.g. Monarch Project at CMU)
- Some things still need work (e.g. integration of mobile IP and 802.11 wireless LANs