PURDUE UNIVERSITY

# Assuring Data Trustworthiness
## *Concepts and Research Challenges*

*Elisa Bertino*

CERIAS and CS Department

Purdue University

CERIAS

Center for Education and Research in Information Assurance and Security

# Motivations

- Data trustworthiness is critical for making “good” decisions
- Few efforts have been devoted to investigate approaches for assessing how trusted the data are
- No techniques exist able to protect against data deception

PURDUE UNIVERSITY

# Approaches

- Integrity models and techniques
  - From the security area:
    - Biba Model
    - Clark-Wilson Model
    - Signature techniques
- Physical integrity
- Semantic integrity
- Data quality
- Reputation techniques

PURDUE UNIVERSITY

# Challenges

Data trustworthiness is a multi-faceted concepts

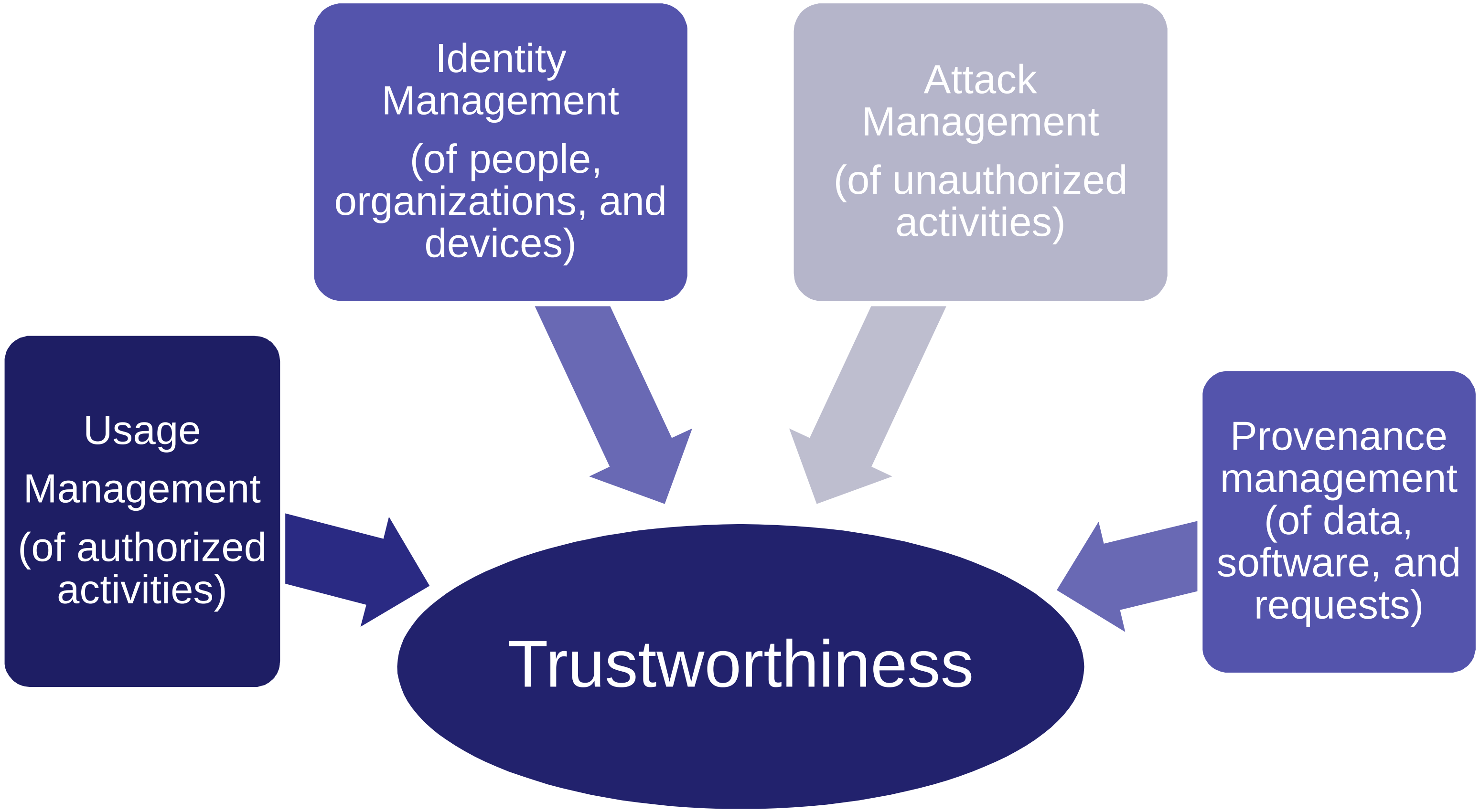
- It means different things to different people or applications
  - The prevention of unauthorized and improper data modification
  - The quality of data
  - The consistency and correctness of data
- Different definitions require different approaches.
  - Access control, workflows, information-flow, constraints, etc.
- We need a unified perspective and approaches to manage and coordinate a variety of mechanisms

# Challenges

The trustworthiness of data is versatile

- It is hard to quantify
- It may change, independent from direct modifications
  - Time, real-world facts
- Its implication may vary, depending on applications
  - High trustworthiness is always preferred
  - However, high trustworthiness often has high costs
- We need flexible systems in which application-dependent policies can be specified and enforced

# The Trust Fabric



# Logical Organization

☞ To assure trustworthiness we need to measure the trustworthiness of identities of people, devices, organizations

☞ The world snapshots are derived, in one way or another, from statements asserted by relevant people, devices, organizations

☞ Usage management seeks to manage authorized activities by extending traditional access control

☞ A usage management system must continuously monitor subjects and data during data accesses by subjects, even after the initial authentication steps

orthiness of some

espect to time as more

☞ *Provenance of data* allows us to measure the trustworthiness of information

☞ *Provenance of software* helps to evaluate the trustworthiness of software programs

☞ *Provenance of requests* enhances the assurance of the requests' source in that they are invoked by the intended subject, rather than by malware

☞ Attack management deals with unauthorized activities, especially malicious attacks

☞ It helps managing the trustworthiness of the infrastructure-level services provided to the other components

# An Example

Data Trustworthiness Assessment
Based on Provenance in Data Streams

# Data Streams Everywhere

- New computing environments
  - Ubiquitous/mobile computing, embedded systems, and sensor networks
- New applications
  - Traffic control systems monitoring data from mobile sensors
  - Location based services (LBSs) based on user's continuously changing location
  - e-healthcare systems monitoring patient medical conditions
  - Real-time financial analysis
- What are we interested in?
  - Data is originated by multiple distributed sources
  - Data is processed by multiple intermediate agents

➡ Assessing **data trustworthiness** is crucial for mission critical applications

➡ Knowing **where the data comes from** is crucial for assessing data trustworthiness

**where the data comes from = Data Provenance**

# What is Provenance?

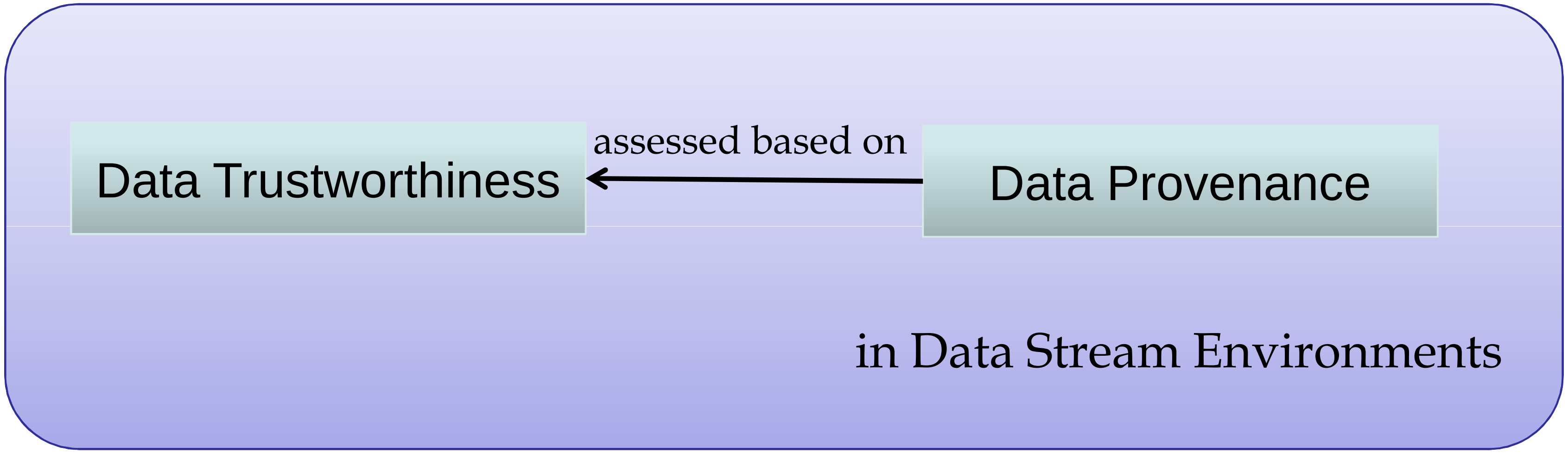
- In general,

  the origin, or history of something is known as its **provenance**.

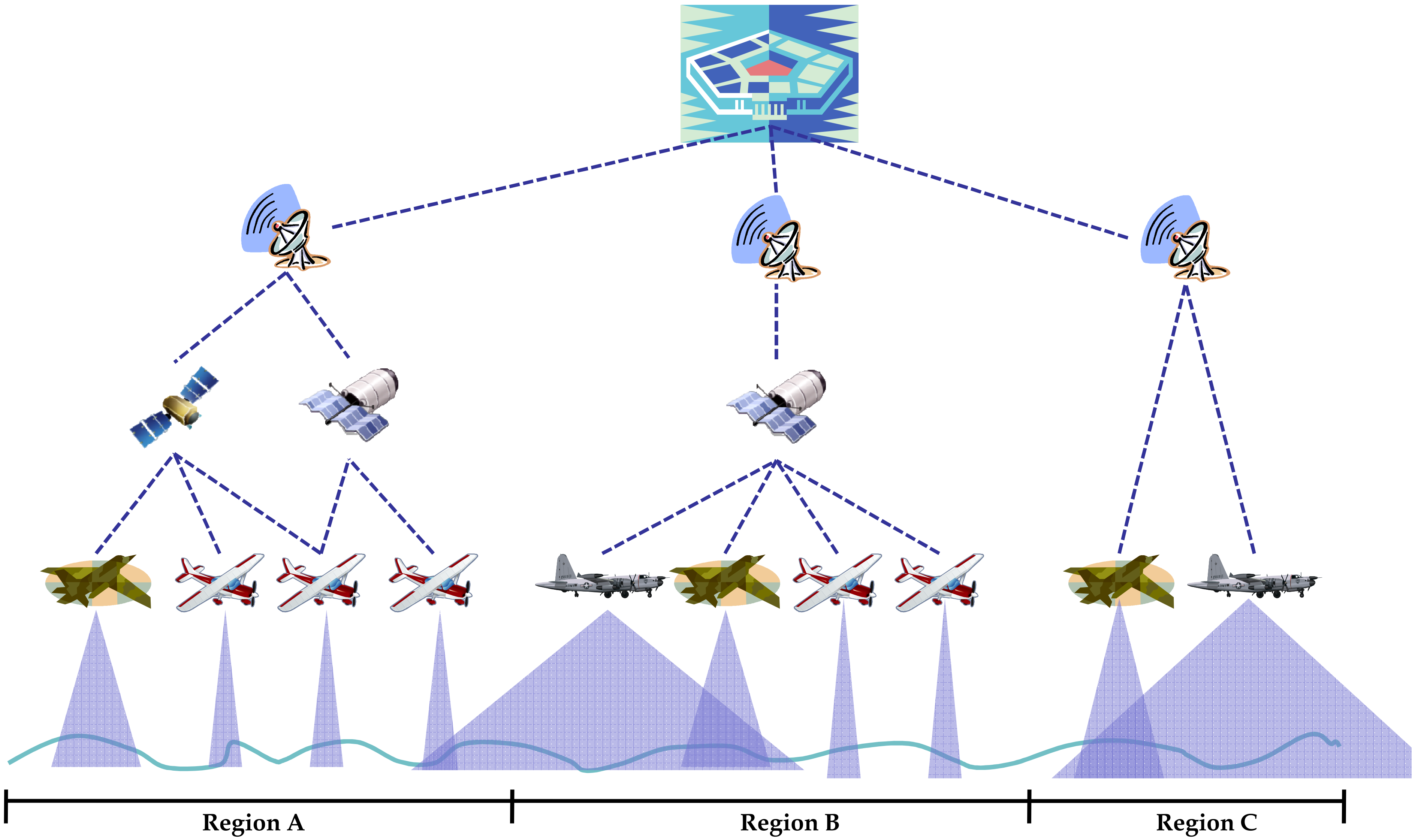
- In the context of computer science,

  **data provenance** refers to information documenting how data came to be in its current state - where it originated, how it was generated, and the manipulations it underwent since its creation.

# Focus of Our Work



# An Example Application: Battlefield Monitoring Sensor Network



# What Makes It Difficult to Solve?

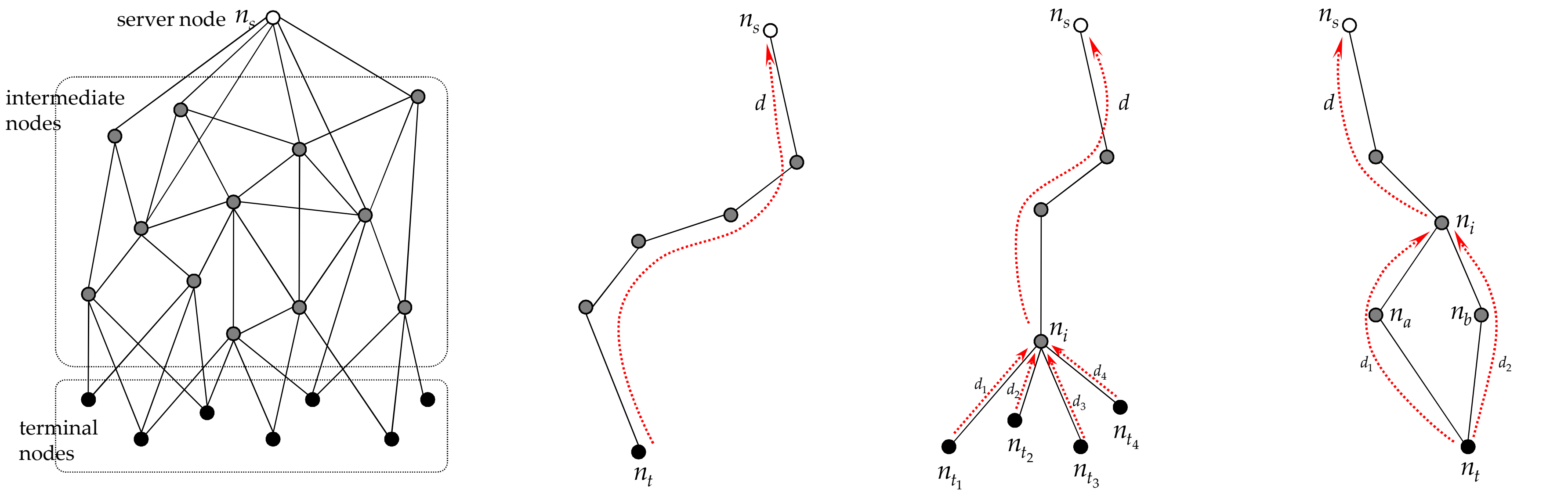
- Data stream nature
  - Data arrives rapidly → real-time processing requirement → high performance processing
  - Unbounded in size → not possible to store the entire set of data items
  - Dynamic/adaptive processing
  - Sometimes, only approximate (or summary) data are available
- Provenance nature
  - Annotation → increased as it is transmitted from the source to the server (i.e., snowballing effect)
  - Interpretation semantics differ from usual data
- Network nature
  - Provenance processing in the intermediate node
    (e.g., provenance information can be merged/separated/manipulated)
  - Hierarchical structure for network and provenance

# Our Solution:

# A Cyclic Framework for Assessing Data Trustworthiness

# Modeling Sensor Networks and Data Provenance

- A sensor network be a graph, $G(N,E)$
  - $N = \{ n_i | n_i$ is a network node of which identifier is $i$ } : a set of sensor nodes
    - a *terminal node* generates a data item and sends it to one or more intermediate or server nodes
    - an *intermediate node* receives data items from terminal or intermediate nodes, and it passes them to intermediate or server nodes
    - a *server node* receives data items and evaluates continuous queries based on those items
  - $E = \{ e_{i,j} | e_{i,j}$ is an edge connecting nodes $n_i$ and $n_j$.} : a set of edges connecting sensor nodes
- A data provenance, $p_d$
  - $p_d$ is a subgraph of $G$



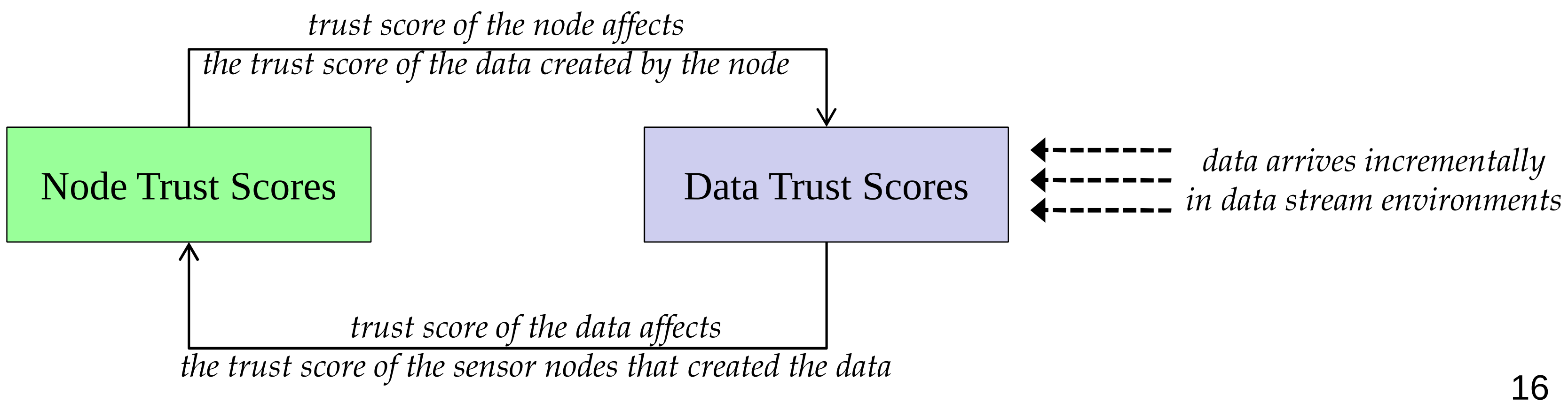

(a) a physical network (b) a simple path (c) an aggregate path (d) an arbitrary graph

# Assessing Trustworthiness → Computing Trust Scores

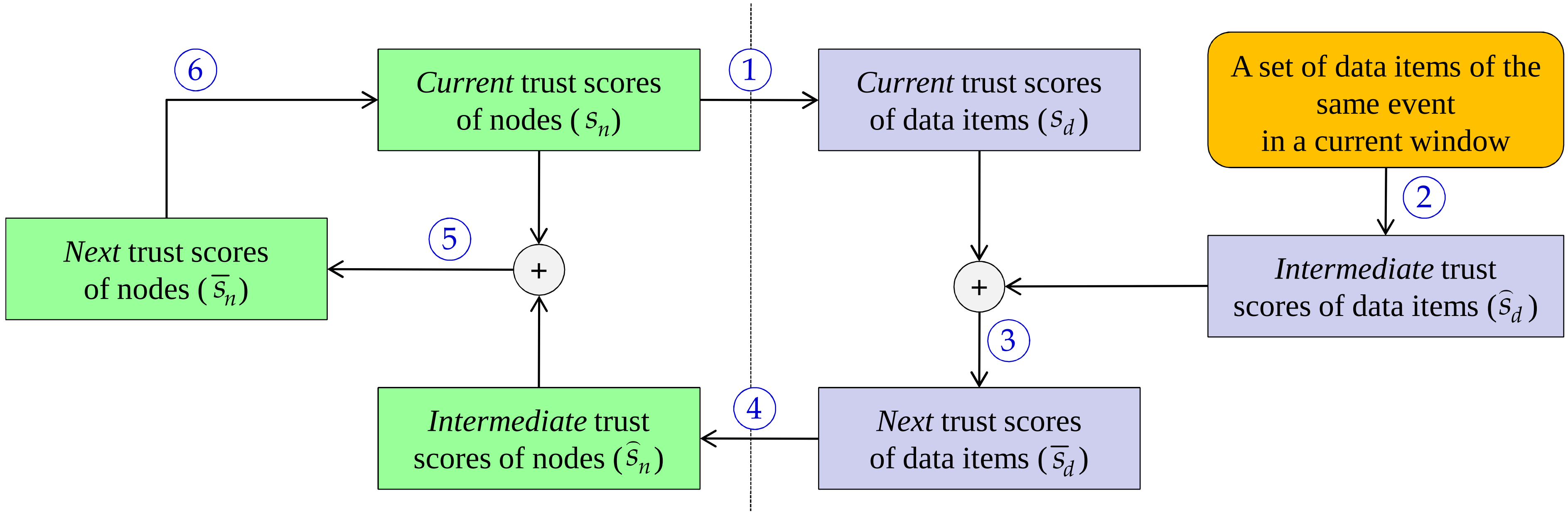
- Trust scores: *quantitative* measures of trustworthiness
  - **Data trust scores**: indicate about how much we can trust the data items
  - **Node trust scores**: indicate about how much we can trust the sensor nodes collect correct data

  ➡ Scores provide an indication about the trustworthiness of data items/sensor nodes and can be used for comparison or ranking purpose

- ***Interdependency*** between data and node trust scores

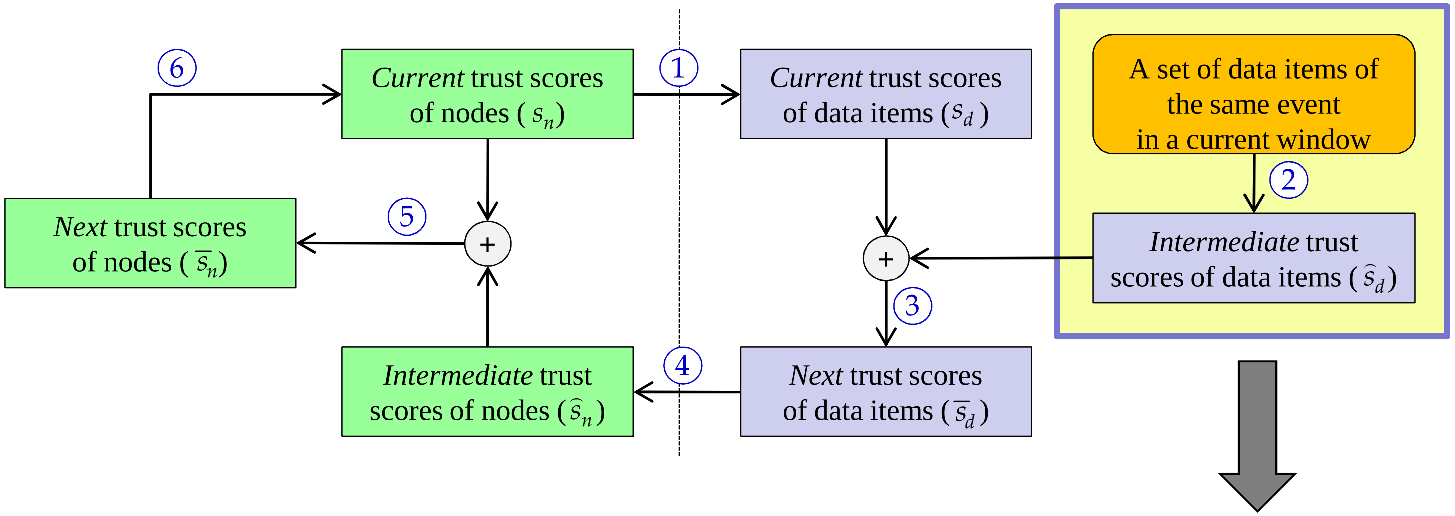


# A Cyclic Framework for Computing Trust Scores




- Trust score of a data item *d*
  - The *current* trust score of *d* is the score computed from the current trust scores of its related nodes.
  - The *intermediate* trust score of *d* is the score computed from a set (d ∈) *D* of data items of the same event.
  - The *next* trust score of *d* is the score computed from its current and intermediate scores.
- Trust score of a sensor node *n*
  - The *intermediate* trust score of *n* is the score computed from the (next) trust scores of data items.
  - The *next* trust score of *n* is the score computed from its current and intermediate scores.
  - The *current* trust score of *n* is the score assigned to that node at the last stage.

# Intermediate Trust Scores of Data (in more detail)




Data trust scores are adjusted according to the ***data value similarities*** and the ***provenance similarities*** of a set of recent data items (i.e., history)

- The more data items have similar values, the higher the trust scores of these items are
- Different provenances of similar data values may increase the trustworthiness of data items

| | Similar Data Value | Different Data Value |
|---|---|---|
| Similar Provenance | score ↑ | score ↓↓↓ (***conflict***) |
| Different Provenance | score ↑↑↑ (***cross checked***) | score ↓ |

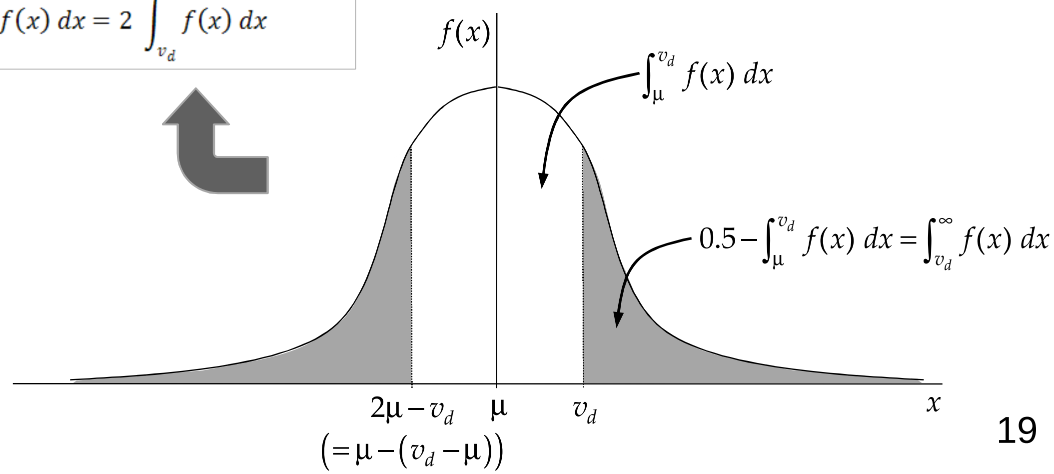
# Using *Data Value* and *Provenance* Similarities

- Setting $\hat{s}_d$ based on data value similarities
  - with the mean $\mu$ and variance $\sigma^2$ of the history data set *D*, we assume the current input data follow a normal distribution N ($\mu$, $\sigma^2$)

a probability density function $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, where $x$ is the value of a data item $d$

  - because the mean $\mu$ is determined by the majority values in *D*,
    - if *x* is close to the mean, it is more similar to the other values;
    - if *x* is far from the mean, it is less similar to the other values.
  - with this observation, we obtain the initial intermediate score of *d* (whose value is $v_d$) as the integral area of *f*(*x*)

$$\text{initial } \hat{s}_n = 2\left(0.5 - \int_{\mu}^{v_d} f(x)\,dx\right) = 1 - \int_{2\mu-v_d}^{v_d} f(x)\,dx = 2\int_{v_d}^{\infty} f(x)\,dx$$



# Using *Data Value* and *Provenance* Similarities (cont'd)

- Adjusting $\hat{s}_d$ with provenance similarities
  - we define the similarity function between two provenances $p_i$, $p_j$ as $sim(p_i, p_j)$
    - $sim(p_i, p_j)$ returns a similarity value in [0, 1]
    - it can be computed from the tree or graph similarity measuring algorithms
  - from the observation of value and provenance similarities, given two data items $d, t \in D$, their values $v_d$, $v_t$, and their provenances $p_d$, $p_t$

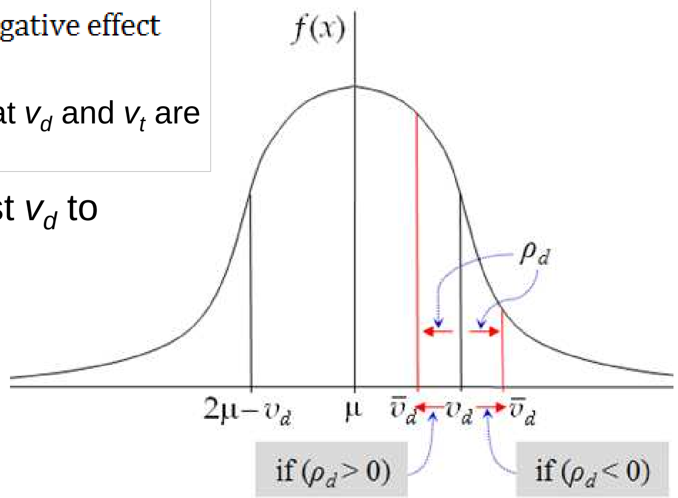
    (here, notation '$\sim$' means "is similar to", and notation '$\not\simeq$' means "is not similar to")
    - if $p_d \sim p_t$ and $v_d \sim v_t$, the provenance similarity makes a small positive effect on $\hat{s}_d$;
    - if $p_d \sim p_t$ and $v_d \not\simeq v_t$, the provenance similarity makes a large negative effect on $\hat{s}_d$;
    - if $p_d \not\simeq p_t$ and $v_d \sim v_t$, the provenance similarity makes a large positive effect on $\hat{s}_d$;
    - if $p_d \not\simeq p_t$ and $v_d \not\simeq v_t$, the provenance similarity makes a small positive effect on $\hat{s}_d$;
  - then, we first calculate the adjustable similarity between $d$ and $t$,

$$\rho_{d,t} = \begin{cases} 1 - sim(p_d, p_t), & \text{if } dist(v_d, v_t) < \delta_1; \quad \text{// positive value and positive effect} \\ -\, sim(p_d, p_t), & \text{if } dist(v_d, v_t) > \delta_2; \quad \text{// negative value and negative effect} \\ 0, & \text{otherwise.} \quad \text{// no effect} \end{cases}$$

where $dist(v_d, v_t)$ is a distance between two values, $\delta_1$ is a threshold that $v_d$ and $v_t$ are treated to be similar; $\delta_2$ is a threshold to be not similar

  - with the (normalized) sum of adjustable similarity of $d$, we adjust $v_d$ to $\bar{v}_d$

$$\rho_d = \sum_{t \in D, t \neq d} \rho_{d,t}$$



# Computing Next Trust Scores

The next trust core is computed as

$$c_d s_d + (1 - c_d)\widehat{s}_d$$

Where is constant ranging in [0,1]

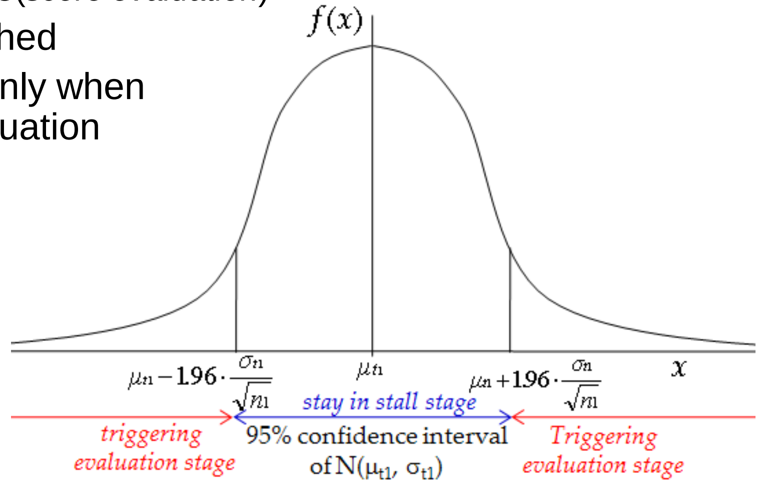
– If is small trust scores evolve fast

– If it large trust scores evolve slowly

– In the experiments we set it to 1/2

# Incremental Evolution of Trust Scores

- Two evolution schemes
  - *Immediate* mode
    - evolves trust scores whenever a new data item arrives
    - pros: provides high accurate trust scores
    - cons: incurs a heavy computation overhead, thus not feasible when the arrival rate of data items is very high
  - *Batch* mode
    - accumulates a certain amount of input data items, and then evolves trust scores only once for the accumulated data items
    - pros: reduces the computation overhead so as to make the cyclic framework scalable over the input rate of data items and the size of sensor networks
    - cons: the accuracy of trust scores can be low compared with the immediate mode
- Batch mode in detail
  - Two stages: Stall Stage(*data accumulation*)/Evolution Stage(*score evaluation*)
  - The evolution stage is triggered when a threshold is reached
  - Use *confidence interval* concept to trigger the evolution only when the current status significantly changed from the last evaluation
    - we use a confidence level $\gamma$ as the threshold
    - trigger only when the mean of accumulated data falls out of the confidence interval of $\gamma$ in the normal distribution of the last evaluation stage
    - an example $\gamma$ = 95%



# Experimental Evaluation

- Simulation
  - Sensor network as an *f*-ary complete tree whose fanout and depth are *f* and *h*, respectively
  - Synthetic data that has a single attribute whose values follow a normal distribution with mean $\mu_i$ and variance $\sigma_i^2$ for each event *i* ($1 \leq i \leq N_{event}$)
  - Data items for an event are generated at $N_{assign}$ leaf nodes and the interval between the assigned nodes is $N_{interleave}$
  - The number of data items in windows (for evaluating intermediate trust scores) is $\omega$

< notation and default values >

| Symbols | Definitions | Default |
|---|---|---|
| $h$ | height of the sensor network | 5 |
| $f$ | fanout of the sensor network | 8 |
| $N_{event}$ | # of unique events | 1000 |
| $N_{assign}$ | # of nodes assigned for an event | 30 |
| $N_{interleave}$ | interleaving factor | 1 |
| $\omega$ | size of window for each event | 20 |

- Goal of the experiments
  - Showing efficiency and effectiveness of our cyclic framework
  - Showing efficiency and effectiveness of batch mode compared to immediate mode

PURDUE UNIVERSITY

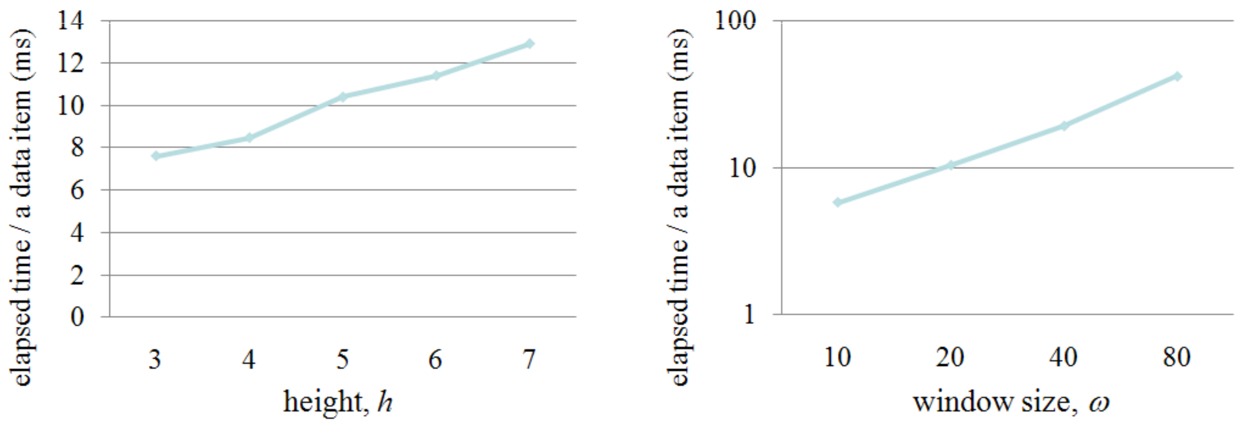
# Experiment 1

## Computation Efficiency of the Cyclic Framework

- Measure the elapsed time for processing a data item with our cyclic framework
- For showing scalability, we varies

  1) the size of sensor networks (i.e., *h*) and

  2) the number of data items for evaluating data trust scores (i.e., *ω*)

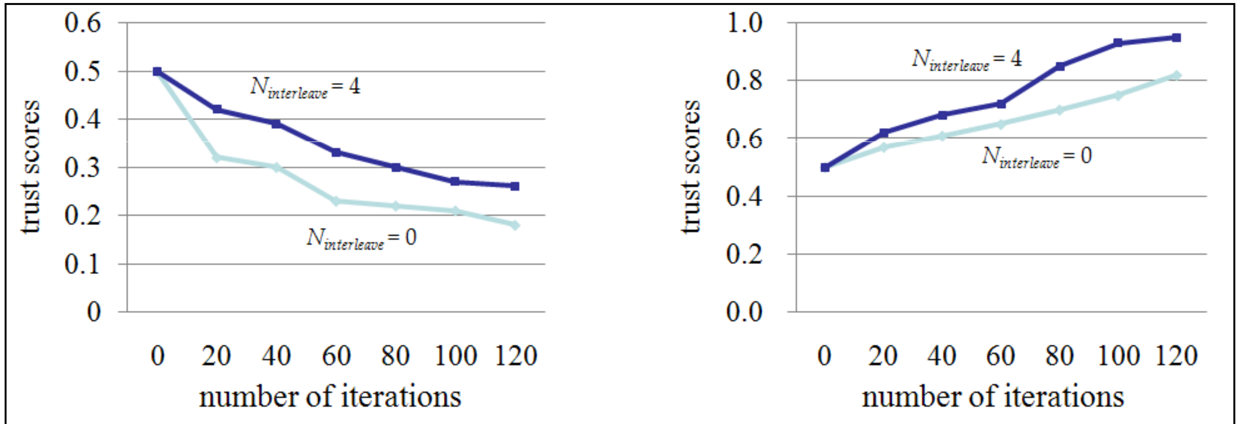



(a) Varying height of the sensor network (b) Varying the window size

- Shows affordable computation overhead and scalability both with the size of sensor network and the number of data items in windows

# Experiment 2
# Effectiveness of the Cyclic Framework

- Inject incorrect data items into the sensor network, and then observed the change of trust scores of data items
- For observing the effect of provenance similarities, we vary the interleaving factor (i.e., $N_{interleave}$) → if $N_{interleave}$ increases, the provenance similarity decreases




(a) With untrustworthy data items (b) With trustworthy data items

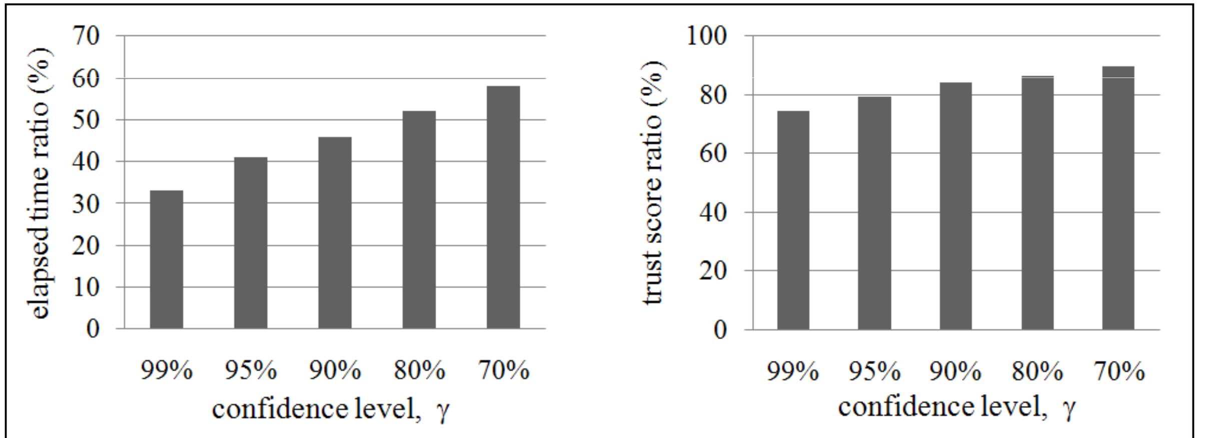
- Graph (a) shows the changes in the trust scores when incorrect data items are injected, and Graph (b) shows when the correct data items are generated again
- In both cases, we can see that our cyclic frame evolves trust scores correctly
- The results also show that our principles
  - *different values with similar provenance result in a large negative effect*
  - *similar values with different provenance result in a large positive effect* are correct

PURDUE UNIVERSITY

# Experiment 3
## Immediate vs. Batch

- Measure the average elapsed time for processing a data item (for efficiency) and measure the average difference of trust scores (for effectiveness)
- For showing the sensitivity on frequency of the evolution stage, we varies the batch threshold (i.e., confidence level $\gamma$)

  → the smaller $\gamma$ means a more frequent invocation of the evolution stage




(a) Efficiency of the batch mode (b) Effectiveness of the batch mode

- From the results, we can see that
  - the performance advantage of the batch mode is high when $\gamma$ is large, and
  - the batch mode does not significantly reduce the accuracy compared with the immediate mode

PURDUE UNIVERSITY

# Discussion

- How do we use trust scores
  - Notion of confidence policy
  - Situation awareness
- How do we improve data assessment
  - Use of semantic knowledge
  - Dynamic integration of new data sources, also heterogeneous
- How do we deal with rapidly changing values
  - User awareness
  - Triggering additional actions, for example collecting more evidence
    - Sensor node sleep/awake times based on data trust scores (required and observed)
- How do we securely convey provenance
  - Data watermarking techniques
- How do we deal with privacy/confidentiality
  - Privacy-preserving data matching techniques

# Another Example

Assessing the Trustworthiness of Location Data Based on Provenance

# Applications and Motivations

- Forensics analysis and disease control
- Locations of individuals (e.g., a suspect was present at the scene of a crime)
- Individuals may lie or information may not be precise
- Mobile computing techniques (GPS, cell phone)
- Approximate information or stolen

# An Example

- Peter's location
  - Chicago, 5pm -> *Lafayette, 8pm -> Cincinnati, 10pm (reported* by a GPS service)
  - Los Angels, 5pm ->*San Francisco, 8pm -> Seattle, 10pm* (reported by a cell phone service)
  - Lafayette, 8pm (reported by the local police)

- Two events are most likely possible: a) Peter was at Lafayette at 8pm; b) Peter was at Seattle at 8pm.

# Problems

– Do the evidence items reported by one source support each other?
– Do the trajectories reported by different sources about an individual support each other?
– Where does the evidence items come from?

PURDUE UNIVERSITY

# Conclusions

- We have started addressing the problem of assessing data trustworthiness based on provenance
- We have proposed initial approaches for sensor networks and location data
- Future work
  - more accurate computation of trust scores
  - secure delivery of provenance information
  - trust scores for aggregation and join in sensor networks
  - extend a streaming data management system with our techniques

# Thank You!

- Questions?
- Elisa Bertino bertino@cs.purdue.edu