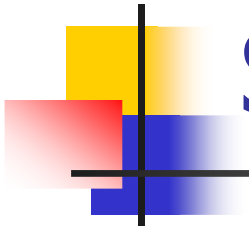
Using Entropy to Trade Privacy for Trust

Yuhui Zhong
Bharat Bhargava
{zhong, bb}@cs.purdue.edu
Department of Computer Sciences
Purdue University

This work is supported by NSF grant IIS-0209059

Problem motivation

- Privacy and trust form an adversarial relationship
  - Internet users worry about revealing personal data. This fear held back $15 billion in online revenue in 2001
  - Users have to provide digital credentials that contain private information in order to build trust in open environments like Internet.
- Research is needed to quantify the tradeoff between privacy and trust.



Sub problems

- How much privacy is lost by disclosing a piece of credential?
- How much does a user benefit from having a higher level of trust?
- How much privacy a user is willing to sacrifice for a certain amount of trust gain?

Proposed approach

- Formulate the privacy-trust tradeoff problem
- Design metrics and algorithms to evaluate the privacy loss. We consider:
  - Information receiver
  - Information usage
  - Information disclosed in the past
- Estimate trust gain due to disclosing a set of credentials
- Develop mechanisms empowering users to trade trust for privacy.
- Design prototype and conduct experimental study

Related work

- Privacy Metrics
  - Anonymity set without accounting for probability distribution [Reiter and Rubin, '99]
  - Differential entropy to measure how well an attacker estimates an attribute value [Agrawal and Aggarwal '01]
- Automated trust negotiation (ATN) [Yu, Winslett, and Seamons, '03]
  - Tradeoff between the length of the negotiation, the amount of information disclosed, and the computation effort
- Trust-based decision making [Wegella *et al.* '03]
  - Trust lifecycle management, with considerations of both trust and risk assessments
- Trading privacy for trust [Seigneur and Jensen, '04]
  - Privacy as the linkability of pieces of evidence to a pseudonym; measured by using *nymity* [Goldberg, thesis, '00]

Formulation of tradeoff problem

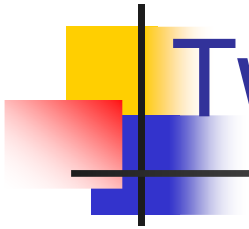
- Set of private attributes that user wants to conceal
- Set of credentials
  - R(i): subset of credentials *revealed* to receiver *i*
  - U(i): credentials *unrevealed* to receiver *i*
- Credential set with minimal privacy loss
  - A subset of credentials *NC* from *U* (i)
  - *NC* satisfies the requirements for trust building
  - PrivacyLoss(*NC*∪*R(i)*) – PrivacyLoss(*R(i)*) is minimized

Formulation tradeoff problem (cont. 1)

- Decision problem:
  - Decide whether trade trust for privacy or not
  - Determine minimal privacy damage
    - Minimal privacy damage is a function of minimal privacy loss, information usage and trustworthiness of information receiver.
  - Compute trust gain
  - Trade privacy for trust if trust gain > minimal privacy damage
- Selection problem:
  - Choose credential set with minimal privacy loss

Formulation tradeoff problem (cont. 2)

- Collusion among information receivers
  - Use a global version R_g instead of *R(i)*
- Minimal privacy loss for multiple private attributes
  - nc_1 better for $attr_1$ but worse for $attr_2$ than nc_2
  - Weight vector $\{w_1, w_2, \ldots, w_m\}$ corresponds to the sensitivity of attributes
    - Salary is more sensitive than favorite TV show
  - Privacy loss can be evaluated using:
    - The weighted sum of privacy loss for all attributes
    - The privacy loss for the attribute with the highest weight



Two types of privacy loss

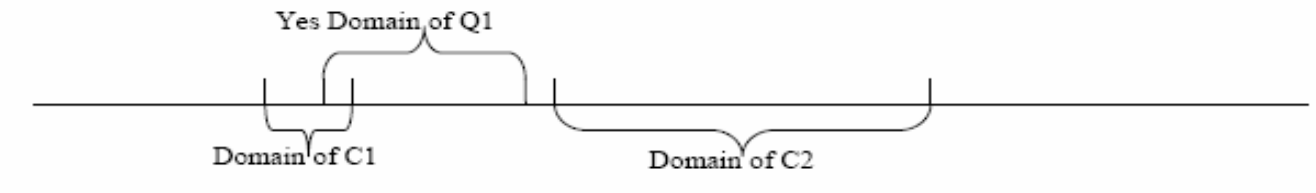
- Query-independent privacy loss
  - User determines her private attributes
  - Query-independent loss characterizes how helpful provided credentials for an adversarial to determine the probability density or probability mass function of a private attribute.

Two types of privacy loss (cont1)

- Query-dependent privacy loss
  - User determines a set of potential queries Q that she is reluctant to answer
  - Provided credentials reveal information of attribute set A. Q is a function of A.
  - Query-dependent loss characterizes how helpful provided credentials for an adversarial to determine the probability density or probability mass function of Q.

Observation 1

- High query-independent loss does not necessarily imply high query-dependent loss
  - An abstract example



Observation 2

- Privacy loss is affected by the order of disclosure
- Example:
  - Private attribute
    - age
  - Potential queries:

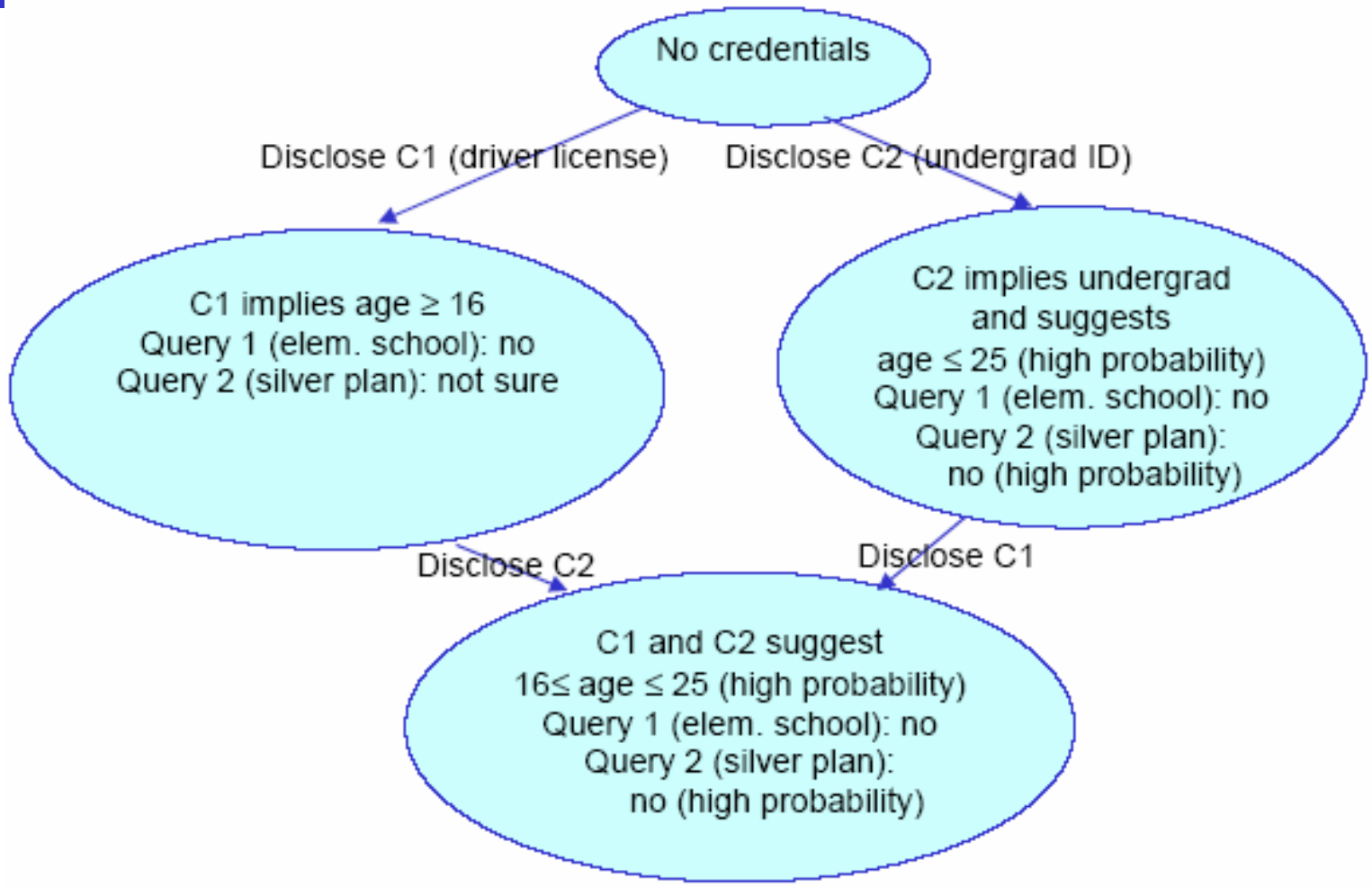
    (Q1) Is Alice an elementary school student?

    (Q2) Is Alice older than 50 to join a silver insurance plan?
  - Credentials

    (C1) Driver license

    (C2) Purdue undergraduate student ID

Example


No credentials
Disclose C1 (driver license)
Disclose C2 (undergrad ID)
C1 implies age ≥ 16
Query 1 (elem. school): no
Query 2 (silver plan): not sure
C2 implies undergrad
and suggests
age ≤ 25 (high probability)
Query 1 (elem. school): no
Query 2 (silver plan):
no (high probability)
Disclose C2
Disclose C1
C1 and C2 suggest
16≤ age ≤ 25 (high probability)
Query 1 (elem. school): no
Query 2 (silver plan):
no (high probability)

Example (cont.)

- C1 → C2
  - Disclosing C1
    - low query-independent loss (wide range for age)
    - 100% loss for Query 1 (elem. school student)
    - low loss for Query 2 (silver plan)
  - Disclosing C2
    - high query-independent loss (narrow range for age)
    - zero loss for Query 1 (because privacy was lost by disclosing license)
    - high loss for Query 2 ("not sure" → "no - high probability"
- C2 → C1
  - Disclosing C2
    - low query-independent loss (wide range for age)
    - 100% loss for Query 1 (elem. school student)
    - high loss for Query 2 (silver plan)
  - Disclosing C1
    - high query-independent loss (narrow range of age)
    - zero loss for Query 1 (because privacy was lost by disclosing ID)
    - zero loss for Query 2

Entropy-based privacy loss

- Entropy measures the randomness, or uncertainty, in private data.
- When an adversarial gains more information, entropy decreases
- The difference shows how much information has been leaked
- Conditional probability is needed for entropy evaluation
  - Bayesian networks, kernel density estimation or subjective estimation can be adopted

Estimation of query-independent privacy loss

- Single attribute
  - Domain of attribute *a*: $\{v_1, v_2, \ldots, v_k\}$

$$PrivacyLoss_a(nc \mid R) = \sum_{i=1}^{k} -P_i \log_2(P_i) - \sum_{i=1}^{k} -P_i^* \log_2(P_i^*)$$

$$where\ P_i = Prob(a = v_i \mid R) \text{ and } P_i^* = Prob(a = v_i \mid R \cup nc)$$

  - P_i and P^*_i are probability mass function before and after disclosing *NC* given revealed credential set R.
- Multiple attributes
  - Attribute set $\{a_1, a_2 \ldots, a_n\}$ with sensitivity vector $\{w_1, w_2, \ldots, w_n\}$

$$PrivacyLoss_A(nc \mid R) = \sum_{i=1}^{n} W_i \times PrivacyLoss_{a_i}(nc \mid R)$$

Estimation of query-dependent privacy loss

- Single query Q
  - Q is the function *f* of attribute set A
  - Domain of *f(A)* : {qv_1, qv_2, …, qv_k}

$$PrivacyLoss_q(nc \mid R) = \sum_{i=1}^{k} -P_i \log_2(P_i) - \sum_{i=1}^{k} -P_i^* \log_2(P_i^*)$$

$$where\ P_i = Prob(f(A) = qv_i \mid R) \text{ and } P_i^* = Prob(f(A) = qv_i \mid R \cup nc)$$

- Multiple queries
  - Query set {q_1, q_2 …,q_n} with sensitivity vector {w_1, w_2, …, w_n}
  - Pr_i is the probability that q_i is asked

$$PrivacyLoss_Q(nc \mid R) = \sum_{i=1}^{n} (PrivacyLoss_{q_i}(nc \mid R) \times Pr_i \times W_i)$$

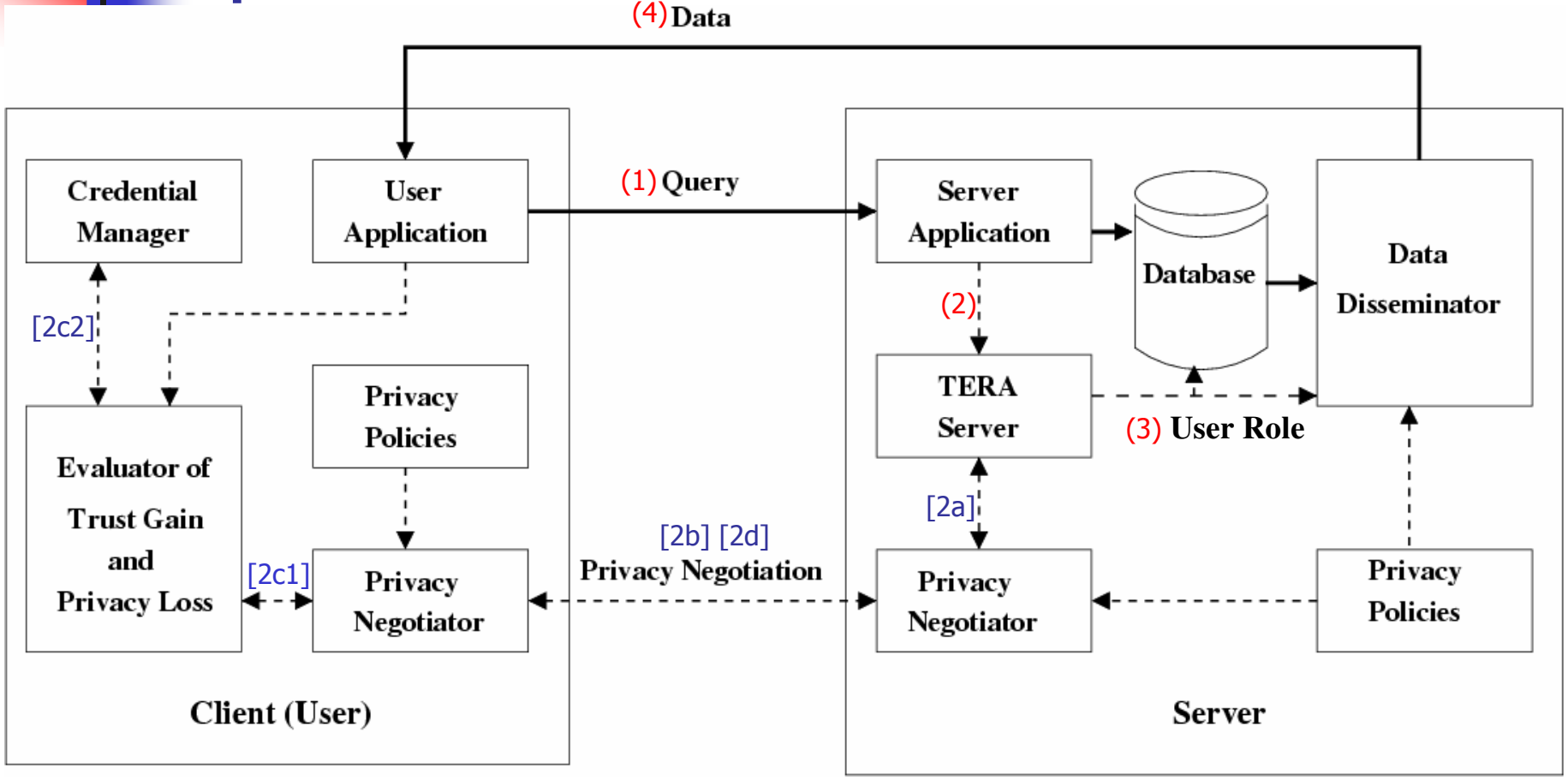
Estimate privacy damage

- Assume user provides one damage function $d_{usage}(PrivacyLoss)$ for each information usage
- PrivacyDamage(PrivacyLoss, Usage, Receiver) = $D_{max}(PrivacyLoss) \times (1\text{-}Trust_{receiver}) + d_{usage}(PrivacyLoss) \times Trust_{receiver}$
  - $Trust_{receiver}$ is a number $\in [0,1]$ representing the trustworthy of information receiver
  - $D_{max}(PrivacyLoss) = Max(d_{usage}(PrivacyLoss)$ for all usage)

Estimate trust gain

- Increasing trust level
  - Adopt research on trust establishment and management
- Benefit function *TB(trust_level)*
  - Provided by service provider or derived from user's utility function
- Trust gain
  - $TB(trust\_level_{new}) - TB(tust\_level_{prev})$

PRETTY: Prototype for Experimental Studies



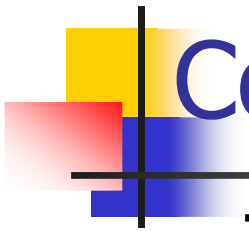
(<nr>) – unconditional path

[<nr>]– conditional path

TERA = Trust-Enhanced Role Assignment

Information flow for PRETTY

1) User application sends query to server application.
2) Server application sends user information to TERA server for trust evaluation and role assignment.
   a) If a higher trust level is required for query, TERA server sends the request for more user's credentials to privacy negotiator.
   b) Based on server's privacy policies and the credential requirements, privacy negotiator interacts with user's privacy negotiator to build a higher level of trust.
   c) Trust gain and privacy loss evaluator selects credentials that will increase trust to the required level with the least privacy loss. Calculation considers credential requirements and credentials disclosed in previous interactions.
   d) According to privacy policies and calculated privacy loss, user's privacy negotiator decides whether or not to supply credentials to the server.
3) Once trust level meets the minimum requirements, appropriate roles are assigned to user for execution of his query.
4) Based on query results, user's trust level and privacy polices, data disseminator determines: (i) whether to distort data and if so to what degree, and (ii) what privacy enforcement metadata should be associated with it.



Conclusion

- This research addresses the tradeoff issues between privacy and trust.
- Tradeoff problems are formally defined.
- An entropy-based approach is proposed to estimate privacy loss.
- A prototype is under development for experimental study.