Data Leakage Detection
AGENDA:

• What is Data Leakage?
• How Data Leakage is done?
• How Data Leakage can be stopped?
INTRODUCTION:

- The main problem to many multinational companies
- Sensitive information being stolen during transmission
- Other organizations both Government and Private facing the problem
- Main trouble causers
EXISTING SYSTEM:

• Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy.
• If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified.
• Watermarks can be very useful in some cases, but again, involve some modification of the original data.
PROPOSED SYSTEM:

- Our goal is to detect when the distributor’s sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data.
- Using the technique of “Perturbation” data is made less sensitive for the agents to handle.
- In this section we develop a model for assessing the “guilt” of agents.
- We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker.
Algorithms:

1. Evaluation of Explicit Data Request Algorithms

2. Evaluation of Sample Data Request Algorithms
Typical Block Diagram Showing the Process of Data Loss In Blocking Spam
Problem Setup and Notation:

A distributor owns a set $T=\{t_1,\ldots,t_m\}$ of valuable data objects. The distributor wants to share some of the objects with a set of agents $U_1,U_2,\ldots,U_n$, but does not wish the objects be leaked to other third parties. The objects in $T$ could be of any type and size, e.g., they could be tuples in a relation, or relations in a database. An agent $U_i$ receives a subset of objects, determined either by a sample request or an explicit request:

1. Sample request
2. Explicit request
Guilt Model Analysis:

Our model parameters interact and to check if the interactions match our intuition, in this section we study two simple scenarios as Impact of Probability $p$ and Impact of Overlap between $R_i$ and $S$. In each scenario we have a target that has obtained all the distributor’s objects, i.e., $T = S$. 
MODULES

1. Data Allocation Module
2. Fake Object Module
3. Optimization Module
4. Data Distributor
Software Required:

O/S : Windows XP.
Language : Asp.Net, c#.
Data Base : Sql Server 2005
Thank You