Anti Inference Hub

A dynamic query processing engine that defends against the Inference Problem in multilevel secure databases

A Work By:
Sari Haj Hussein

in fulfillment of Frontiers of Programming Language Course
Secure and Dependable Computer Systems Master's Programme
Chalmers University of Technology
7/5/2010

Table of Contents

What is the Inference Problem?

- An Inference Channel is a construction by which an attacker can deduce sensitive data from nonsensitive data.
- The Inference Problem is the problem of identifying and then removing any inference channel in a database.

Why is the Inference Problem Difficult?

- A database is open to queries.
- When queries' results are put together, we learn about new data.
- Thus, it is difficult to determine the exact amount of data that can be learnt.

Remainder: MLS and SDB

- MLS = Multilevel Secure Databases
  - The security of one element may be different than the security of other elements in the same row or column.
  - We need several levels of security in a database; only sensitive and nonsensitive are not enough.
- SDB = Statistical Databases
  - They are Online Analytical Processing (OLAP) systems.
  - They enable users to retrieve only aggregate statistics (count, sum, average or standard deviation).
  - They are used as data warehouses or data mines for the purpose of business intelligence.
- Anti Inference Hub is centric around MLS.

Example Inference Problems

- We will cover the following examples:
  - Inference from Queries Based on Sensitive Data
  - Inference in Statistical Databases
  - Inference from Key Integrity
  - Inference from Functional and Multivalued Dependencies
  - Inference from Value Constraints

Inference from Queries Based on Sensitive Data


```
SELECT commanders.name
FROM commanders, missions
WHERE commanders.mission = missions.code;
```

$\pi_{commanders.name}\sigma_{commanders.mission = missions.code}(commanders \times missions)$

- We have an inference channel! Why?
- Cause: sensitive data are used to create the Cartesian product.
- Remedy: modify the query or abort its execution.

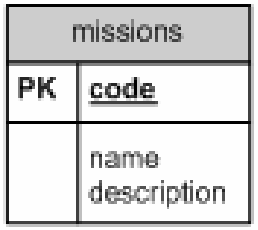
Inference in Statistical Databases

- A hospital database that stores patients' medical records of the form {Age, Sex, Employer, Social Security Number, Diagnosis Type}.
- Physicians can access everything.
- Researchers can only perform aggregations e.g. COUNT [(Sex = Male) & (Employer = Volvo)].
- An evil researcher wants to illegally determine the diagnosis type of a patient Lisbeth whose age is 34 and works for Ericsson. Is it possible you think?

Inference in Statistical Databases

- Yes it is!
- Query 1: COUNT [(Age = 34) & (Sex = Female) & (Employer = Ericsson)]
- Think of the consequences if Query 1 returns 0 or 1!
- Query 2: COUNT [(Age = 34) & (Sex = Female) & (Employer = Ericsson) & (Diagnosis Type = Insomnia)]
- Think of the consequences if Query 2 returns 0 or 1!

Inference from Key Integrity



classification	code	name	description
sensitive data	0XX	Skyscraper	Move the artillery to Ohio

classification	code	name	description
nonsensitive data	0XX	Dogscratcher	Move the artillery to Missouri

- We want to preserve key integrity!
- What are the options we have?

Inference from Key Integrity

- Solution: Polyinstantiation!

classification	code	name	description
sensitive data	0XX	Skyscraper	Move the artillery to Ohio
nonsensitive data	0XX	Dogscratcher	Move the artillery to Missouri

Inference from Functional and Multivalued Dependencies

Classification: Nonsensitive Data
Classification: Nonsensitive Data
Classification: Sensitive Data

salaries	
PK	**empname**
	emprank empsalary

- Same rank means same salary.
- We have an inference channel! Why?
- Cause: emprank → empsalary.
- Remedy: raise the classification level of emprank to sensitive data.

Inference from Value Constraints

Classification: Nonsensitive Data
Classification: Nonsensitive Data
Classification: Nonsensitive Data
Classification: Sensitive Data

items	
PK	**code**
	name cost price

- We have the value constraint price – cost ≤ 1500.
- We have an inference channel! Why?
- Cause: value constraint is defined over several sensitivity levels.
- Remedy: partition the value constraint.

Techniques for Dealing with Inference Channels

- Semantic data modeling techniques: search an entire database for illegal information flow, then give advice on how to redesign the database to avoid the flow.
  - High false positive rate - identifying an inference channel when it is not inference channel.
  - High false negative rate - missing an inference channel when it is an inference channel.
- Query analysis techniques: analyze queries dynamically and block queries that lead to inference.

Semantic Data Modeling or Query Analysis?

- Query analysis techniques are favored over semantic data modeling techniques for two main reasons:
  - Evaluating a query dynamically is less expensive than searching an entire database for possible information flow.
  - Data is constantly added to (or updated in) a database. This may open up new inference channels that cannot be identified other than dynamically.
- Still some suffer of slow query processing time.
- Anti Inference Hub is based on a query analysis technique.

Staddon's Novel Approach [2003]

- The first dynamic query analysis technique that does not largely slow down query processing time.
- Staddon's technique assumes that inference channels have already been identified at pre-query processing time.
- C-collusion resistance meaning that a coalition of c users cannot together query all objects in an inference channel (we call c the degree of collusion resistance).
- Crowd control meaning that even if a coalition of users have queried all but one object in an inference channel, none of them will be able to query the remaining object.

Staddon's Steps

1) Key allocation: allocate a key set for each user.
2) Database initialization: allocate a token set for each object in an inference channel (how and why?).
3) Dynamic query processing: if a token t in T_i is used to gain access to object O_i, then for every $s \neq i$, any token in T_s that was generated using the same key is deleted.

Staddon's Steps Illustrated

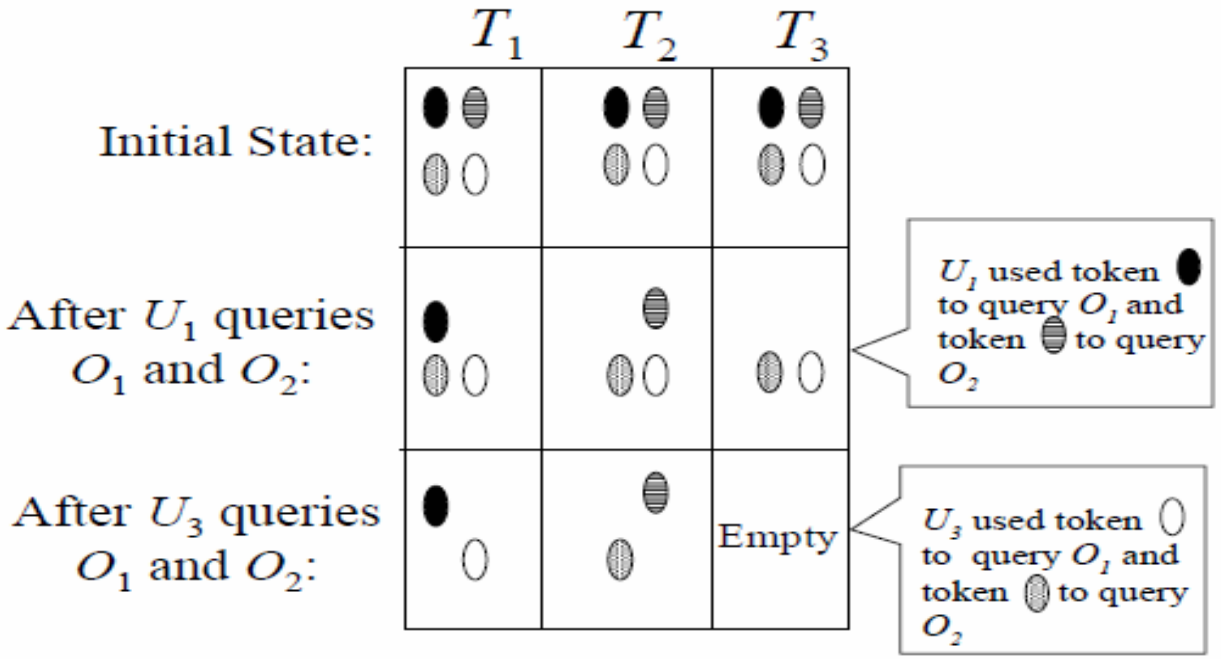
Inference Channel of Length 3: $\{O_1, O_2, O_3\}$

4 Users:

U_1's Tokens = ● ◍ U_3's Tokens = ○ ◍

U_2's Tokens = ● ◍ U_4's Tokens = ○ ◍

Dynamic Inference Control:



Chen-Wei Refinement 2005

- The most efficient dynamic query analysis presented so far.
- Chen-Wei's technique also assumes that inference channels have already been identified at pre-query processing time.
- Two kinds of key schemes:
  - The key set is only used by the database system so that users do not need to keep any keys.
  - Each user has one secret key.

Chen-Wei Steps

1) Key initialization: associations between keys and objects are established; runs one time.
2) Query processing: details the algorithm of a query; runs whenever a user wants to access an object.

Chen-Wei Single Key Set Schemes

- Chen-Wei presented three single key set schemes:
  - Single Inference Channel
  - Multiple Inference Channels Without “Repeated Objects”.
  - Multiple Inference Channels With “Repeated Objects”.

Single Inference Channel

Key initialization:

$K(O_i) = K, i = 1, \cdots, m.$

Query processing:

Input: i;

if $K(O_i) = \emptyset$ **then**

output "access denied";

else

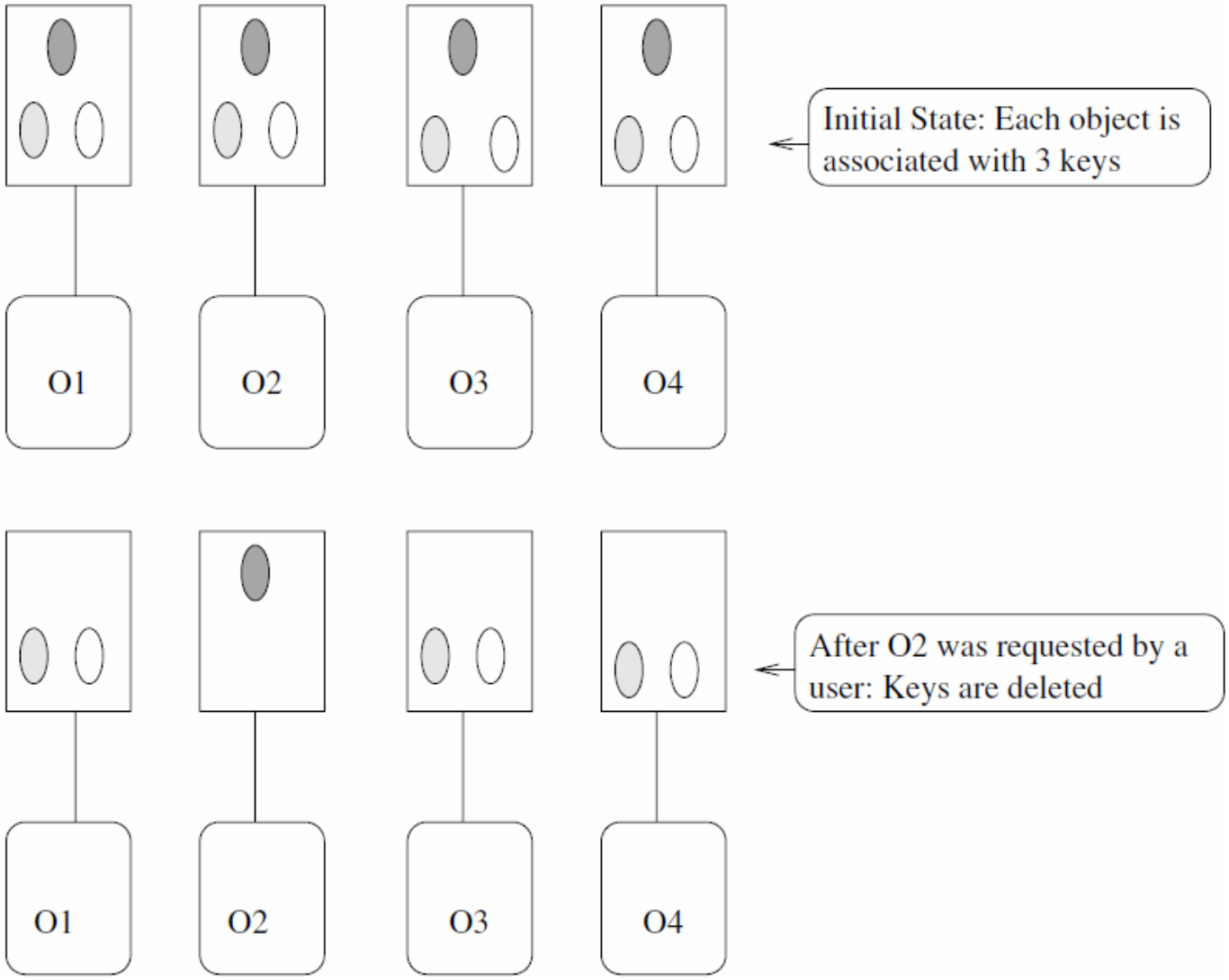
Select randomly a $k_j \in K(O_i)$;

$K(O_i) = \{k_j\}$;

$K(O_s) = K(O_s) \backslash \{k_j\}$ for all $s \neq i$;

Deliver O_i to the user.

Single Inference Channel



Multiple Inference Channels With and Without "Repeated Objects"

- Similar approach; however, we allocate one key set for each inference channel.
- We should pay attention to synchronization across inference channels!
- Once a repeated object is indicated as a reserved object, we should make it the reserved object of all other channels in which it appears.

Drawback of Chen-Wei Single Key Set Schemes

- Block-an-Object Attack: a form of DoS attack in which a malicious user visits all the m − 1 other objects in an inference channel so that the last object is blocked (reserved object).
- Remedy: the database administrator defines Super Clients who are allowed to access reserved objects and make inference.

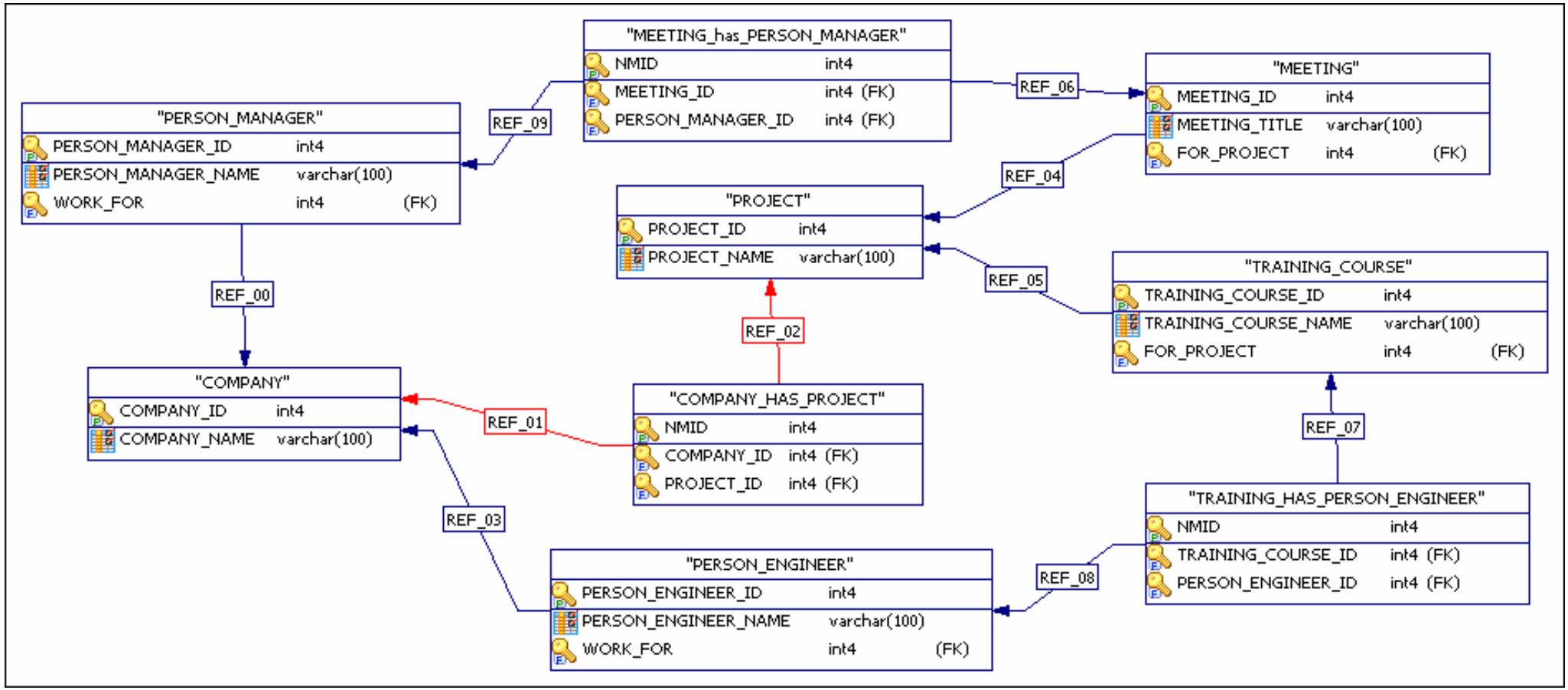
Chen-Wei Surpasses Staddon

	Staddon's Scheme	Our Scheme
Cost	1. A list of acceptable key for each object 2. Each user has $(m-1)/c$ keys 3. Mechanism to prevent against key leaks	1. System tables
Parameters	1. Processing time: depending on m and q 2. Access flexibility:$[1,(m-1)/c]$ 3. Key space: $q(m-1)/c$ 4. Key size: large 5. Collusion resistance: $\leq c$	1. Processing time: depending on m 2. Access flexibility: $m-1$ 3. Key space: $m-1$ 4. Key size: small 5. Collusion resistance: Any size

Testing Anti Inference Hub

- Anti Inference Hub is an implementation of Chen-Wei refinement.
- Let’s thwart an inference attempt using the Hub!

The Sample Database



The Inference Attempt to Address

- Suppose that a low user is able to know the following by executing queries against the database:
  - For which COMPANY a PERSON_MANAGER works.
  - PERSON_MANAGER attending a MEETING.
  - MEETING on a PROJECT.
- If that was true, then the low user can immediately infer the COMPANY supporting the PROJECT.
- Anti Inference Hub can be used to thwart this inference attempt.

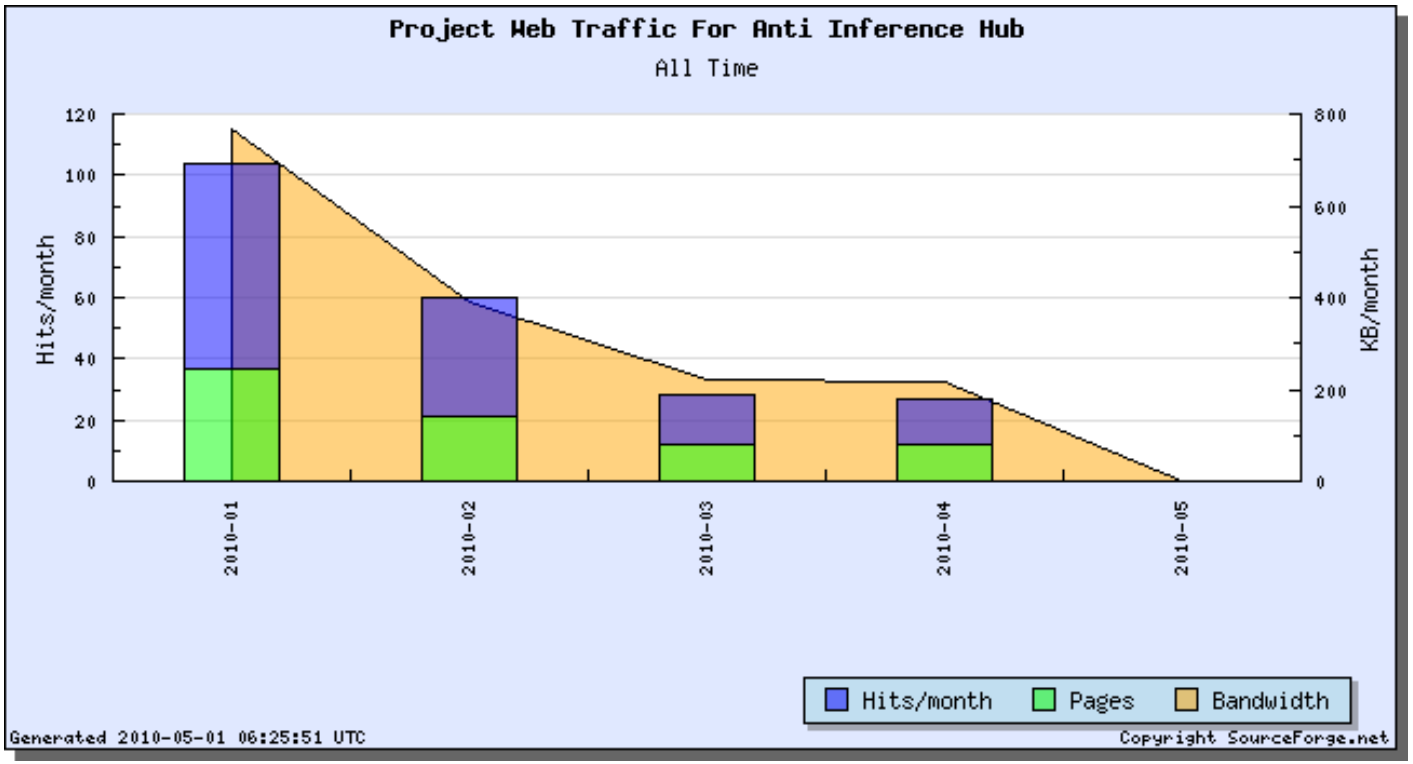
Difficulties Encountered

- Implementing Chen-Wei Multiple Inference Channels With “Repeated Objects” scheme was a highly complicated task.
- We had to find an efficient method to generate unique keys for Chen-Wei. Lastly we used the **U**niversally **U**nique **ID**entifier (java.util.UUID) first shipped with Java 5.0.
- Some networking issues were really problematic.
- Staddon, Chen and Wei did not respond to any of our enquiries!

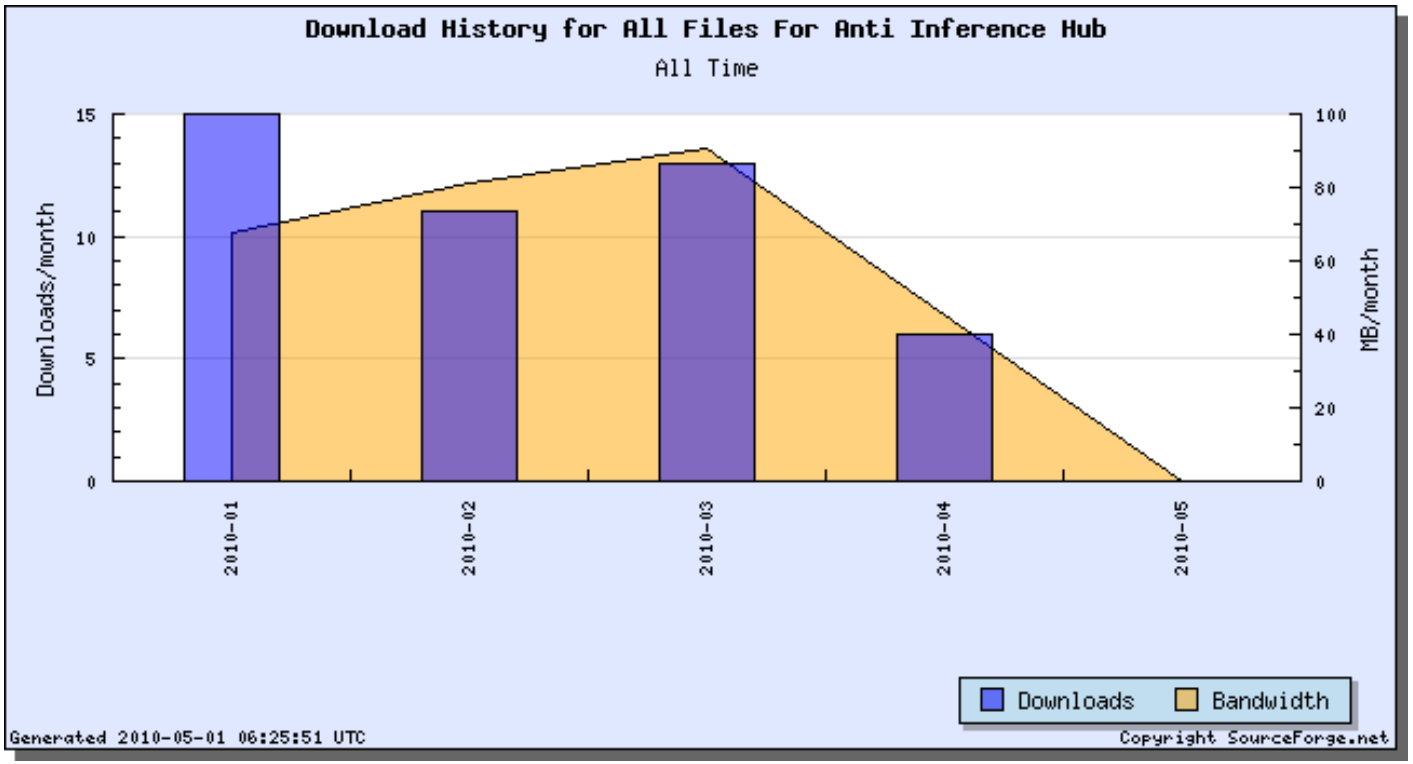
Future Work

- Anti Inference Hub will automatically (and as accurately as possible) locate inference channels in a database.
- Anti Inference Hub will secure statistical databases as well.
- Anti Inference Hub will be pluggable into web servers e.g. Apache.
- More workforce is needed.

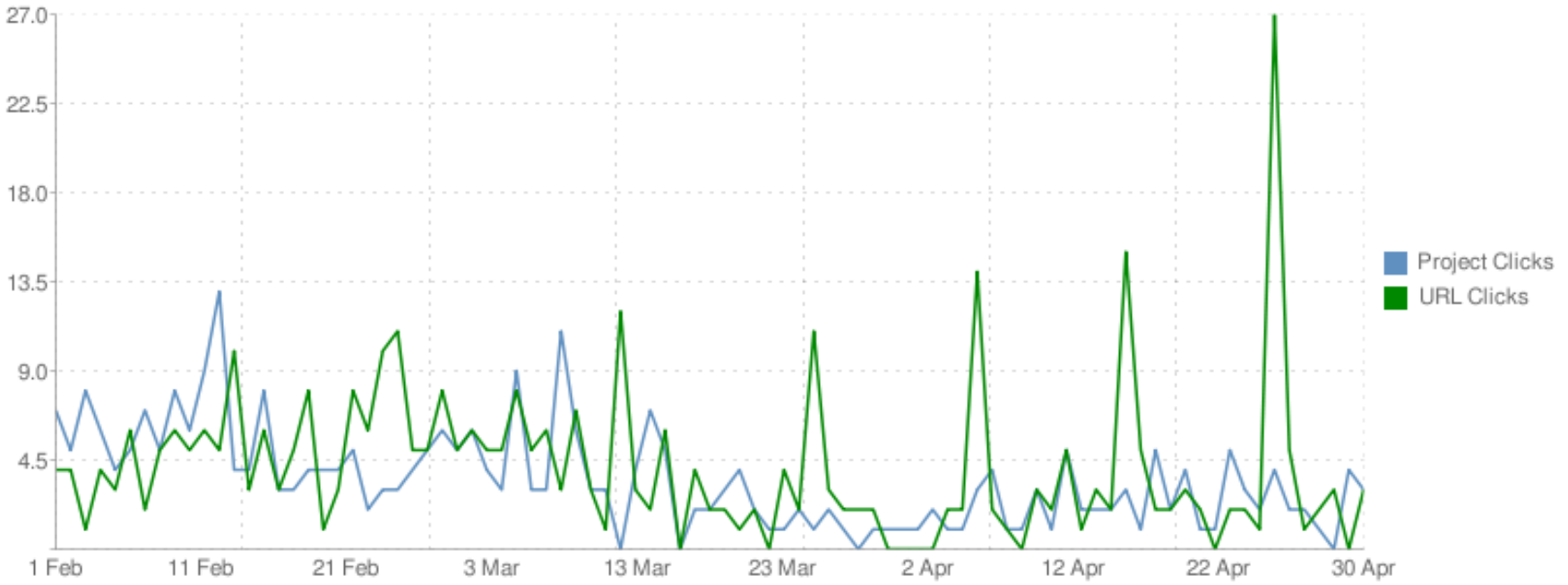
Anti Inference Hub on Sourceforge (http://sourceforge.net/projects/aih/)



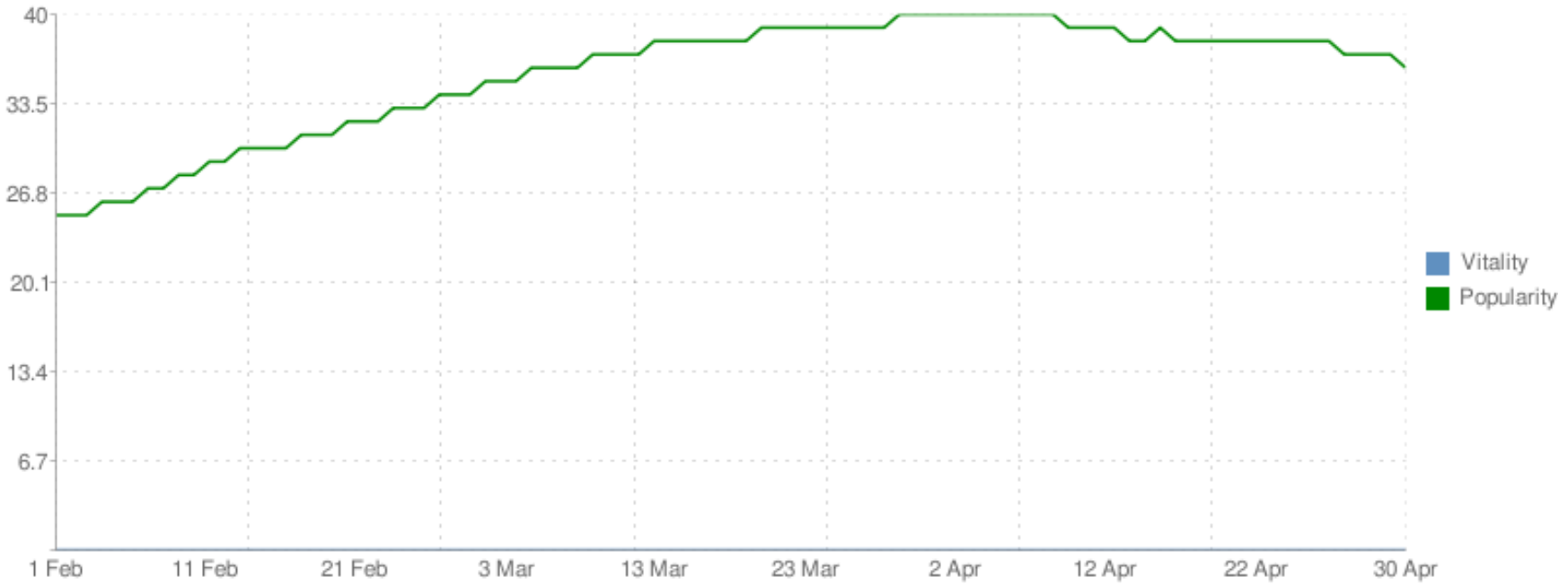
Anti Inference Hub on Sourceforge (http://sourceforge.net/projects/aih/)



Anti Inference Hub on Freshmeat (http://unix.freshmeat.net/projects/anti-inference-hub)



Anti Inference Hub on Freshmeat (http://unix.freshmeat.net/projects/anti-inference-hub)



References

1) Sushil Jajodia, and Catherine Meadows, "Inference Problems in Multilevel Secure Database Management Systems," in "Information Security: An Integrated Collection of Essays,"IEEE Computer Society Press, Los Alamitos, CA USA, 1995.
2) Charles P. Pfleeger, and Shari Lawrence Pfleeger, "Security in Computing, Fourth Edition," Prentice Hall, 2006.
3) Adam, Nabil R., and John C. Wortmann, "Security-Control Methods for Statistical Databases: A Comparative Study," ACM Computing Surveys, Vol. 21, No. 4, Dec. 1989, pp. 515-556.
4) Staddon, Jessica, "Dynamic Inference Control," Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, 2003.
5) X. Chen, and R. Wei, "Dynamic Method for Handling the Inference Problem in Multilevel Secure Databases," Proceedings of the International Conference on Information Technology: Coding and Computing, 2005.

Questions please!