Security in Outsourcing of Association Rule Mining

Wai Kit Wong, David Cheung, Ben Kao and Nikos Mamoulis,

The University of Hong Kong

Edward Hung, *The Hong Kong Polytechnic University*

VLDB 2007, Vienna, Austria

Agenda

- Introduction and motivation
- Item mapping and encryption
- The algorithm for valid and complete transaction transformation
- Experiments
- Summary

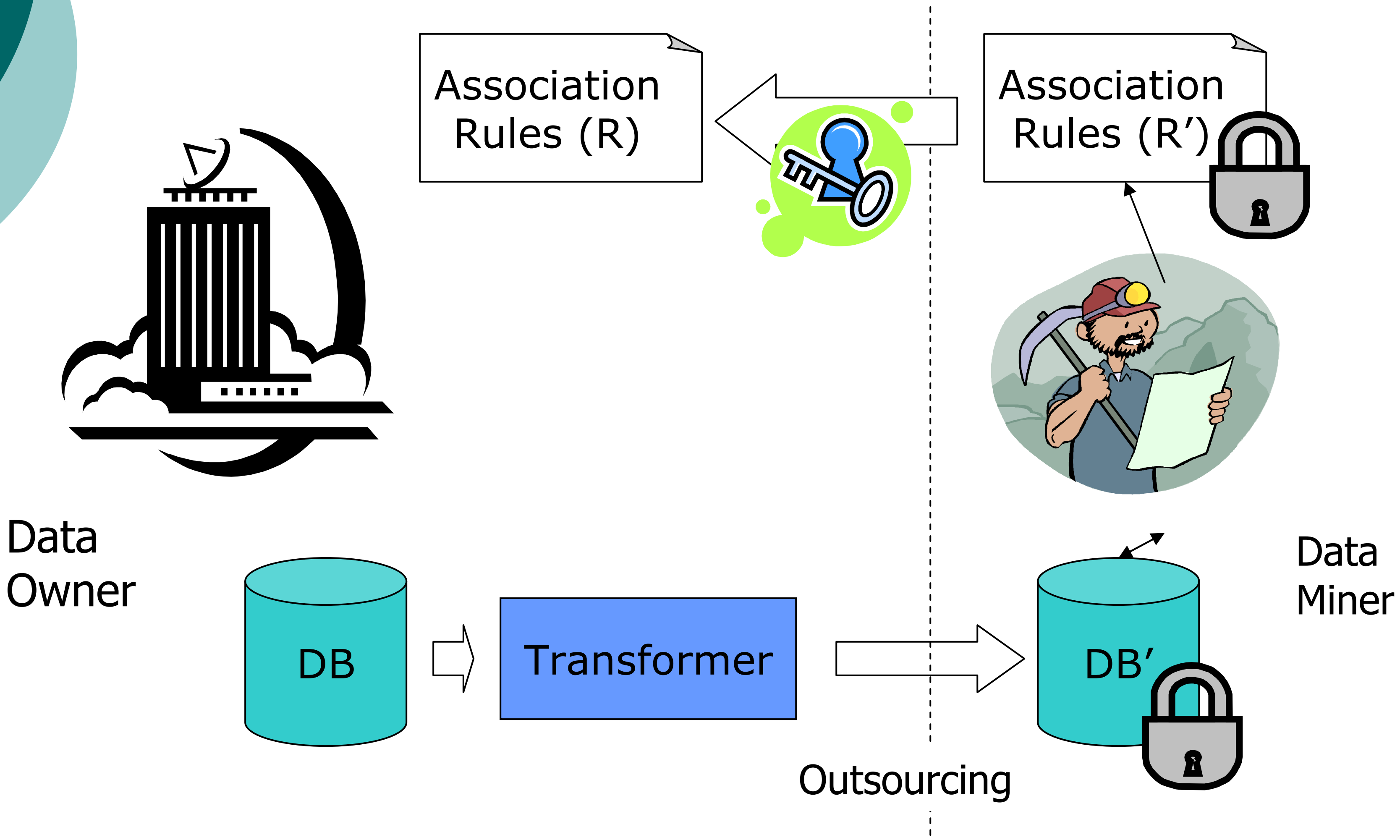
Introduction and motivation

- Association rule mining
  - complexity of exponential order
- Motivation on outsourcing of mining task
  - lower cost
  - avoid hiring in-house specialists
  - consolidate data from different sources

Security concerns in outsourcing

- The third party cannot be trusted
- Need to protect
  - Protect the input – prevent the miner (third party) to access the original transaction records
  - Protect the output – prevent the miner to see the “true” association rules

Outsource model



Item mapping - encryption

Example item mapping (one-to-one)

- bread -> 54
- chocolate -> 165
  - <bread, chocolate> -> <165, 54>
- <54, 165> is large to the miner
  - <cheese, book> or <bread, chocolate>?
- Similar to substitution cipher used in encryption of text
- Anything more secure ????

One-to-n item mapping

- A one-to-n item mapping
  - B: a set of items
  - m: I -> 2^B
- Example, I = {a,b,c},

  B = {1,2,3,4,5}
  - m(a) = {1, 4, 5}
  - m(b) = {2}
  - m(c) = {3, 5}
- Is one-to-n more secure ?

Itemset mapping using one-to-n item mapping

- m: I -> 2^B : one-to-n item mapping
- M: 2^I -> 2^B : itemset mapping

- $M(X) = U_{x \text{ in } X}\ m(x) = Y$
- $M^{-1}(Y) = X$, if $M(X) = Y$
- Example:
  - M(<a, c>) = <1, 3, 4, 5>
  - M(<b, c>) = <2, 3, 5>

  - M^{-1}(<1, 3, 4, 5>) = <a, c>
  - M^{-1}(<1, 2, 3, 4, 5>) = <a, b, c>

m:
a -> {1, 4, 5}
b -> {2}
c -> {3, 5}

- Note: m is an item mapping, M is the itemset mapping

Correctness – restrictions on one-to-n mapping

m:
a -> {1, 2}
b -> {2, 3}
c -> {1, 3}

- <a, b>=><1, 2, 3>
- <a, b, c>=><1, 2, 3>

Collisions!

Decryption failure!

m′:
a -> {1, 2}
b -> {2, 3}
c -> {2, 4}

- <a>=><1, 2>

…

- <a,b>=><1, 2, 3>

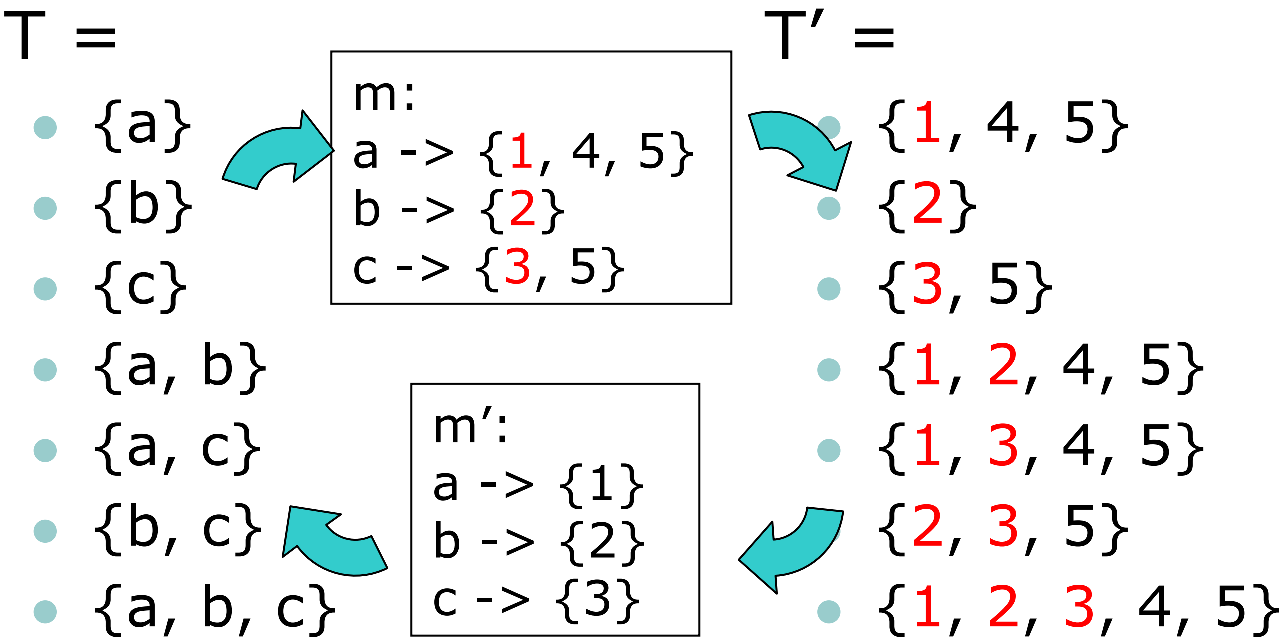
…

- <a, b, c>=><1, 2, 3, 4>

Admissiable Mapping : mapping of each item contains a unique item

Result : $M^{-1}(M(X)) = X$ (correct decryption) iff m is admissible

Is one-to-n mapping more secure?



T =

- {a}
- {b}
- {c}
- {a, b}
- {a, c}
- {b, c}
- {a, b, c}

m:
a -> {1, 4, 5}
b -> {2}
c -> {3, 5}

T' =

- {1, 4, 5}
- {2}
- {3, 5}
- {1, 2, 4, 5}
- {1, 3, 4, 5}
- {2, 3, 5}
- {1, 2, 3, 4, 5}

m':
a -> {1}
b -> {2}
c -> {3}

To decrypt transactions encrypted by **m**, we can use **m'**!

(m is not more secure than m') !!!!

Function coverage

- M_1: 2^I -> 2^{D1}
- M_2: 2^I -> 2^{D2}
- M_1 covers M_2 iff
  - for all X □ I, let $Y = M_2(X)$
    - $M_2^{-1}(Y) = M_1^{-1}(Y \cap D1)$
- M_1 covers M_2
  - If any transaction encrypted by M_2 can be decrypted by using the inverse of M_1

One-to-n is not more secure than one-to-one mapping

- Our results (proved)
  - Any admissible one-to-n itemset mapping is covered by (can be decrypted by) some one-to-one itemset mapping
- Bad news !!!
  - One-to-n item mapping is NOT more secure than a one-to-one item mapping

One-to-n vs one-to-one

- one-to-n vs one-to-one?
  - Intuitively, one-to-n should be more secure

Unfortunate Scenario:

- one-to-n + item mapping

  = one-to-one + item mapping

Our solution :

- Add a random component to transaction transformation
- It will make one-to-n always better (more secure) than one-to-one

One-to-n Transformation

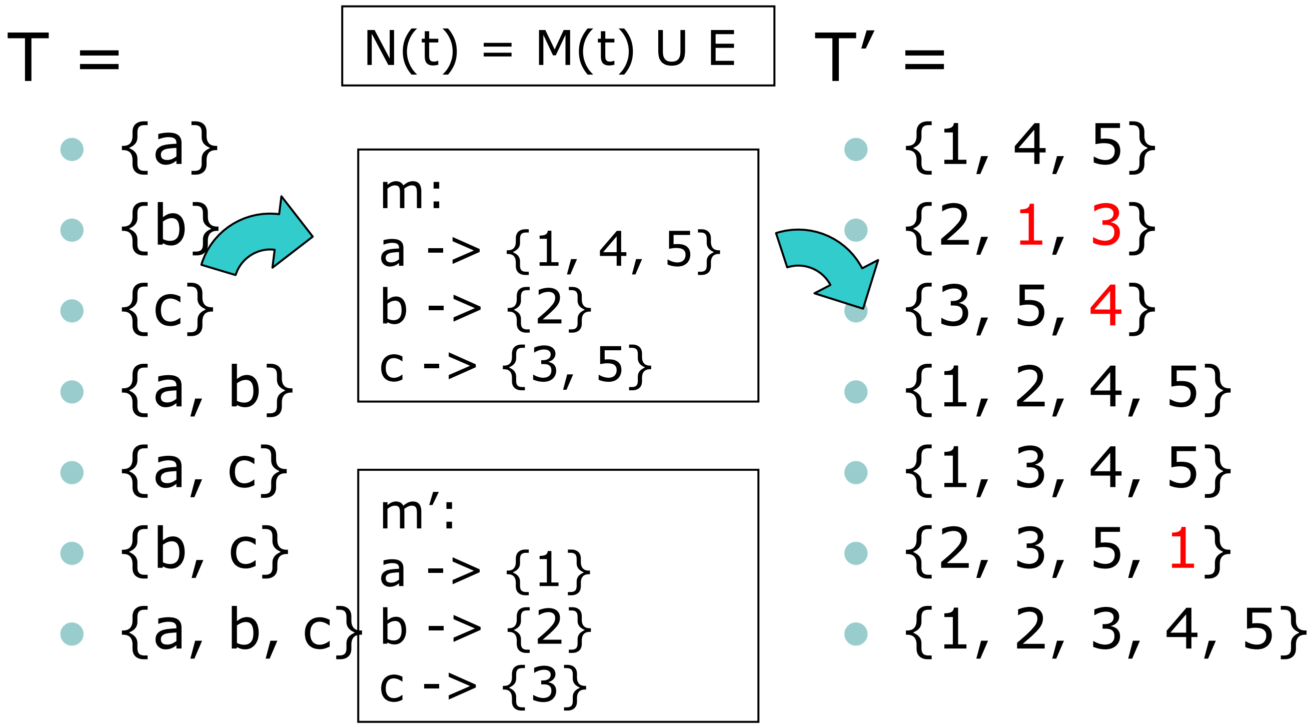
- one-to-one mapping
  - a -> { 1 }, b -> { 2 }, …
  - t = { a, b } → t′ = { 1, 2 }
- one-to-n mapping
  - a -> { 1, 3 }, b -> { 2, 3 }, …
  - t = { a, b } → t′ = { 1, 2, 3 }
- one-to-n transformation
  - a -> { 1, 3 }, b -> { 2, 3 }, …
  - t = { a, b } → t′ = { 1, 2, 3, 4, 6 }

Randomly generated

Transaction transformation

- M: 2^I -> 2^B , based on a one-to-n itemset mapping m
- N: transaction transformation
  - Maps from 2^I to 2^{BUF}
- t' = N(t) = M(t) U E
  - E is a random subset of B U F; F is a set of items not in B
- N^{-1}(t') = {x | m(x) in t'}

Example transformation



N(t) = M(t) U E

T =

- {a}
- {b}
- {c}
- {a, b}
- {a, c}
- {b, c}
- {a, b, c}

m:
a -> {1, 4, 5}
b -> {2}
c -> {3, 5}

m’:
a -> {1}
b -> {2}
c -> {3}

T’ =

- {1, 4, 5}
- {2, 1, 3}
- {3, 5, 4}
- {1, 2, 4, 5}
- {1, 3, 4, 5}
- {2, 3, 5, 1}
- {1, 2, 3, 4, 5}

- The randomly inserted values does not affect the correctness of the decryption

- m’ can no longer be used to decrypt m !!

Necessary properties of transformation N

- Valid
  - The decryption is correct
  - $N^{-1}(N(t)) = t$
- Complete (based on valid)
  - For every transaction t, N(t) generates every possible t’ (= M(t) U E) such that $N^{-1}(t') = t$
- Positive result : No one-to-one itemset mapping can cover a valid and complete transaction transformation from a one-to-n itemset mapping

Generating E for valid and complete transformation N

E = ?

m:
a -> {1, 4, 5}
b -> {2}
c -> {3, 5}

N(<c>) = <3, 5> U E

- For m: a -> {1, 4, 5}

  E = {1} or {4}, but not {1, 4}

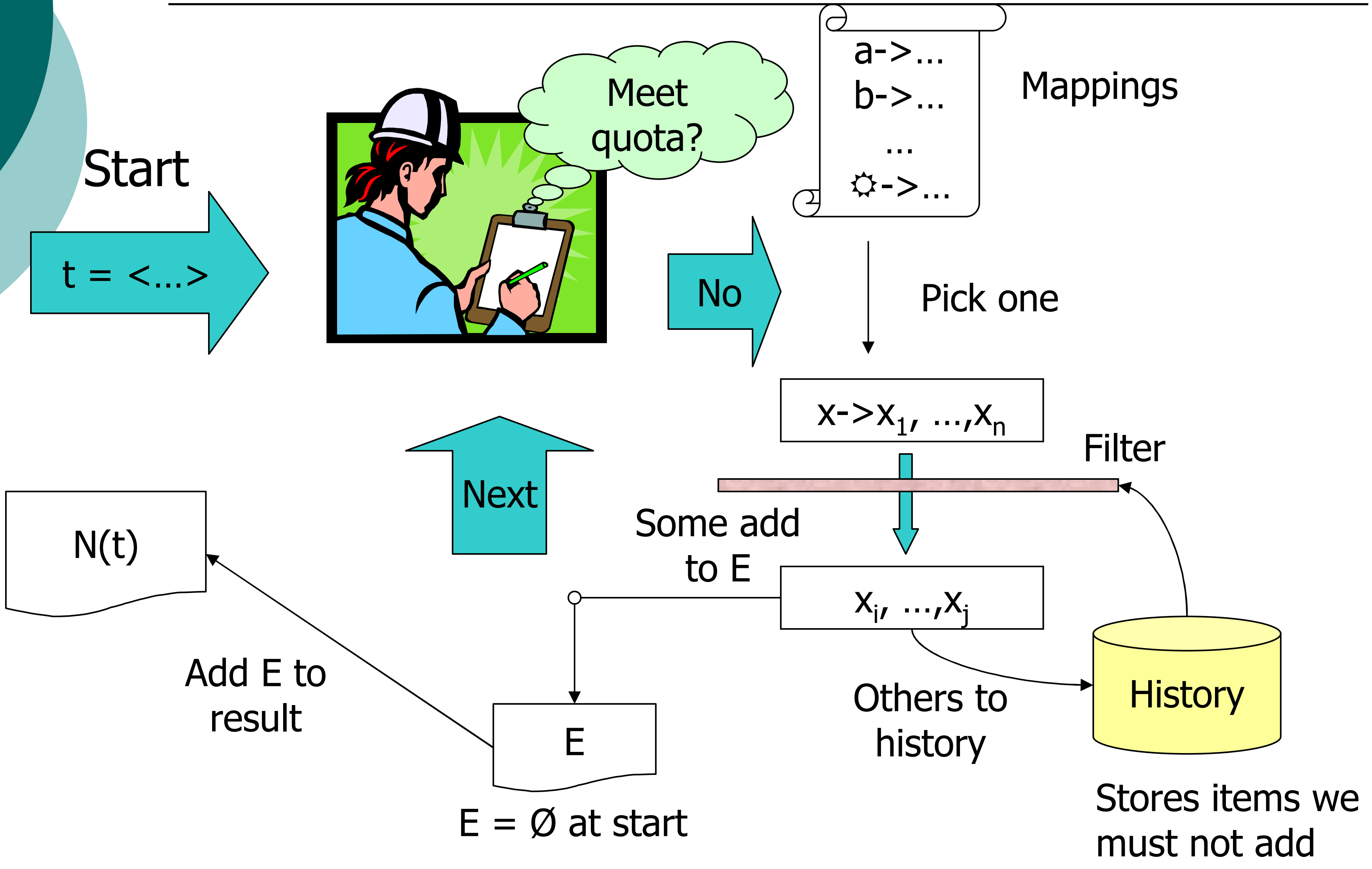
- For m: b -> {2}

  E = Φ

- The transformation N is valid if E is either {1} or {4} or Φ ;
- N is complete if it is possible to generate all of the three cases, i.e., E = {1} or {4} or Φ.

Algorithm – valid and complete transaction transformation

Algorithm to perform valid and complete transformation



Start

t = <…>

Meet quota?

No

a->…
b->…
…
☼->…

Mappings

Pick one

$x \text{->} x_1, \ldots, x_n$

Filter

Some add to E

$x_i, \ldots, x_j$

Others to history

History

Stores items we must not add

Next

N(t)

Add E to result

E

E = Ø at start

Important Property

- The transaction transformation produced by the Algorithm is valid and complete.

Experiments

Design

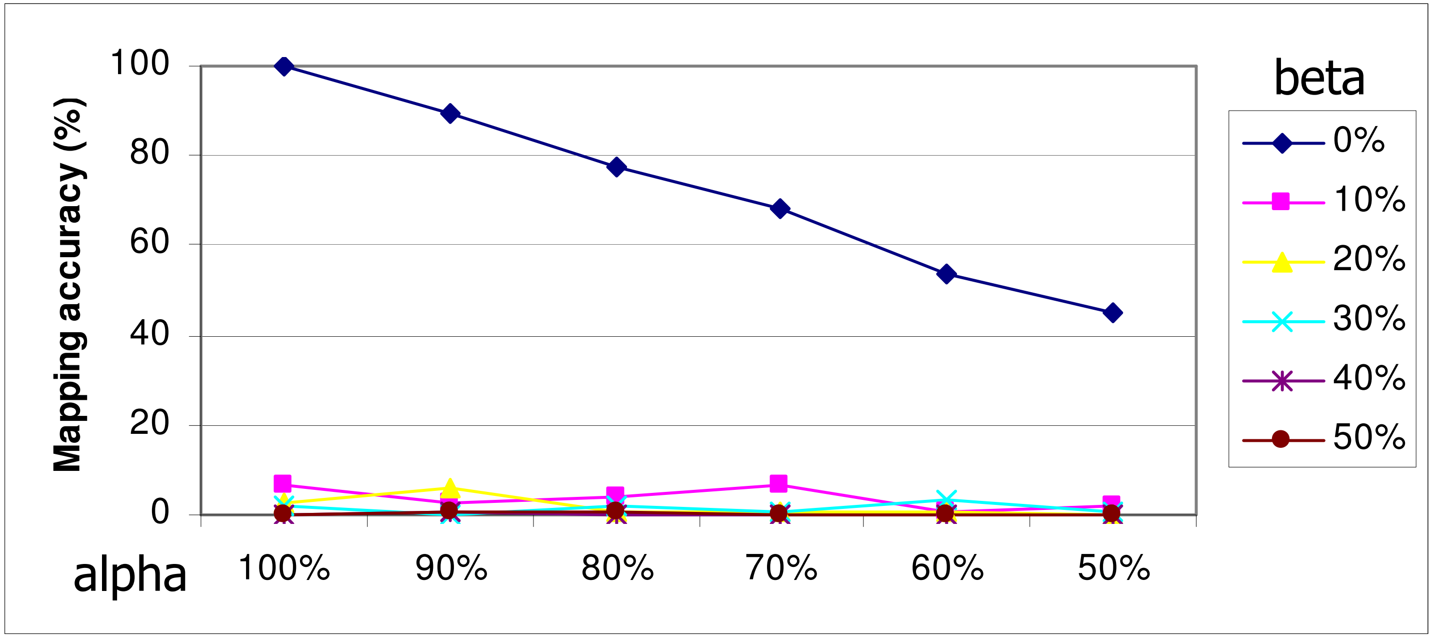
- Purpose
  - Study security and efficiency of the model
- Security
  - Assume the attacker gets the relative frequencies
  - Implemented genetic algorithm for frequency analysis
- Efficiency
  - Transformation time vs mining time
  - Overhead at the miner side

Background knowledge

- Purpose: simulate a real attacker in practice
- Where does the attacker get knowledge? (Assumption)
  - In many cases, the statistics of the global industry is public (background knowledge)
- Background Knowledge (with two parameters)
  - alpha: knows alpha% of large itemsets in original database
  - beta: the support in the knowledge is in the range
    - real support * (1 ± beta)

Mapping accuracy

- Measure how many mapping is correct
  - Only measure those in background knowledge since there is no info for other mappings



Efficiency

	100k	200k	300k	400k	500k
Cost at owner side (transformation and recovery)	2.8s	5.5s	9.5s	11.2s	12.5s
Cost at miner side	195s	488s	738s	945s	1122s
Original mining cost	80s	204s	293s	383s	465s

Summary

- The idea of substitution cipher is used in the problem of encryption of transaction database
- One-to-n item mapping cannot be directly applied since it is effectively a one-to-one item mapping
- Transaction transformation is proposed and shown to be valid and complete
- Experiments show that it is suitable for outsourcing

End