



Tamper Detection and Localization for Categorical Data Using Fragile Watermarks

Yingjiu Li¹, Huiping Guo² and Sushil Jajodia²

¹Signapore Management University

²Center for Secure Information System
George Mason University

Oct. 25, 2004

Outline

- Introduction
- Related work
- Tamper Detection and Localization for Categorical Data Using Fragile Watermarks
- Security analysis
- Conclusion

Introduction

- Applications
 - Database outsourcing
 - Data owners may not have enough resources
 - They outsource their data management needs to an external service provider.
 - The service provider hosts their databases and provides database service to clients
 - Edge computing
 - The central server put its databases on edge servers that are close to clients
 - To save bandwidth and to facilitate process of queries from clients
- Problem
 - The third party servers are not trusted
 - Check integrity of database relations

Related work: watermarking databases

- **Robust watermarking for databases**
 - R. Agrawal, VLDB'02; R.Sion, SIGMOD'03
 - Copyright protection for databases
 - Only watermarking numerical attributes
- **Why fragile watermarking for databases**
 - Detect modifications made to a database relation
 - Traditional digital signature or MAC
 - Can only detect the modifications
 - Can not locate the modifications
 - Need extra space

Tamper Detection and Localization for Categorical Data Using Fragile Watermarks

■ Motivation

- Check integrity of database relations which are put in un-trusted servers

■ Advantages

- Detect and locate modifications
- Applicable to database relation with categorical attributes
- Do not need extra space to store authentication information

■ Basic approach

- Embed invisible watermarks directly to database relations in a distortion free way

Watermark embedding

- Overview

- Grouping

- All tuples are securely divided into groups

- Watermark constructing

- For each group, a unique watermark is constructed

- Watermark embedding

- The order of tuples in a group is modified to represent the embedded watermark

Grouping

- Primary key hash

$$h_i^P = \text{HASH}(K, r_i.P)$$

- Grouping is secret

$$k = h_i^P \bmod g$$

$$r_i \rightarrow \mathcal{G}_k$$

- Ideal hash function $h(M)$

- One way
- $h(M') \neq h(M)$
- Any tamper of the input will randomize the output

P	A1	A2	A3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Embed a watermark to a group

- Tuple hash

$$h_i = \text{HASH}(K, r_i.A_1, r_i.A_2, \dots, r_i.A_\gamma)$$

- Group hash

$$\mathcal{H} = \text{HASH}(K, h_1, h_2, \dots, h_\nu)$$

- Watermark construct

$$W = \text{extractBits}(\mathcal{H}, \nu/2)$$

The length of W : $\nu/2$

The number of tuples: ν

The number of tuple pairs: $\nu/2$

- Embed the watermark

- A watermark bit W_i is embedded into a tuple pair $\langle r_i, r_{i+1} \rangle$

- Embedding rule

$$W_i = 1$$

the first tuple hash > the second tuple hash

$$W_i = 0$$

the first tuple hash < the second tuple hash

- Switch $\langle r_i, r_{i+1} \rangle$ if needed
- Embedding is distortion free

Watermark detection

- Grouping
 - Need to know K and g
 - Same as that in watermark embedding
- Watermark construct W
 - From group hash
 - W is supposed to be embedded
- Watermark extract W'
 - From the tuple order
- Watermark verification
 - $W == W'$?
- Localization
 - group

P	A1	A2	A3
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Watermarking a small table

0	Paul
1	Marry
2	Tom
3	Joe
4	Lucy
5	Bob
6	Jack
7	Mike
8	Cathy
9	Mona

Table (g=2)

1009
2001
1005
4310
1000
2357
2100
1111
3294
3000

Tuple hash

Group 1
Wm1 = {01}

0	Tom
2	Paul
3	Joe
6	Jack

1005
1009
4310
2100

} 0
} 1

Group 2
Wm2 = {101}

1	Marry
4	Lucy
5	Mike
7	Bob
8	Cathy
9	Mona

2001
1000
1111
2357
3294
3000

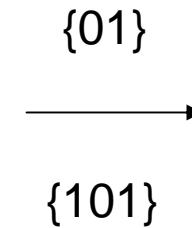
} 1
} 0
} 1

Watermarked table

0	Paul
1	Marry
2	Tom
3	Joe
4	Lucy
5	Bob
6	Jack
7	Mike
8	Cathy
9	Mona

Before embedding

1009
2001
1005
4310
1000
2357
2100
1111
3294
3000



Watermarks

0	Tom
1	Marry
2	Paul
3	Joe
4	Lucy
5	Mike
6	Jack
7	Bob
8	Cathy
9	Mona

After embedding

1005
2001
1009
4310
1000
1111
2100
2357
3294
3000

The watermarked table is tampered

0	Tom
1	Marry
2	Paul
3	Joe
4	Lucy
5	Mike
6	Jack
7	Bob
8	Cathy
9	Mona

Watermarked table

0	Tyler
1	Marry
2	Paul
3	Joe
4	Lucy
5	Mike
6	Jack
7	Bob
8	Cathy
9	Mona

Tampered table

1705
2001
1009
4310
1000
1111
2100
2357
3294
3000

Tuple hash

1

Detect the modification: case 1

Group 1 Wm1 = {10}	0	Tyler	1705	"1"	X	1
	2	Paul	1009			
	3	Joe	4310	"1"		
	6	Jack	2100			
Group 2 Wm2 = {101}	1	Marry	2001	✓		
	4	Lucy	1000			
	5	Mike	1111			
	7	Bob	2357			
	8	Cathy	3294			
	9	Mona	3000			

Detect the modification: case 2

Group 1
Wm1 = {0}

2	Paul	1009
3	Joe	4310
6	Jack	2100

} "1" X

Group 2
Wm2 = {110}

0	Tyler	1705
1	Marry	2001
4	Lucy	1000
5	Mike	1111
7	Bob	2357
8	Cathy	3294
9	Mona	3000

} "0" X
} "0"
} "0"

Security analysis

- Two parameters
 - The number of attributes γ
 - The number of tuples in a group ν
- Probability
 - All *alterations* are correctly localized in corresponding groups
- Simple *alterations*
 - Modify an attribute value; insert a tuple; delete a tuple
- Massive *alterations*
 - Modify multiple attribute values; insert multiple tuples; delete multiple tuples

Modify an attribute value

- Non-primary key attribute
 - Grouping unaffected
 - Modification randomizes group watermark of length $v/2$
 - Probability of modification being correctly detected is $prob=1-1/2^{(v/2)}$
- Primary key attribute
 - Modified tuple has prob $1/g$ to stay, and prob $1-1/g$ to move
 - Prob of modification being correctly detected is →

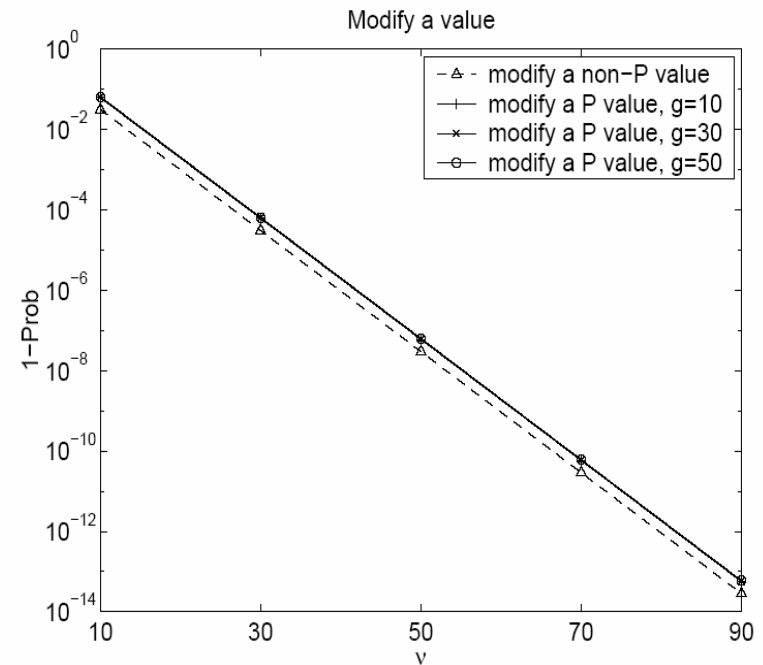


Figure 2: Error in detecting single value modification

$$Prob = \frac{1}{g} \left(1 - \frac{1}{2^{v/2}}\right) + \frac{g-1}{g} \left(1 - \frac{1}{2^{v/2}}\right) \left(1 - \frac{1}{2^{v/2}}\right)$$

Insert/delete a tuple

- Insert/delete a tuple
 - increases/decreases group watermark length
 - randomizes the affected watermark
- Comparison
 - The longer the watermark, the lower the error rate in detection

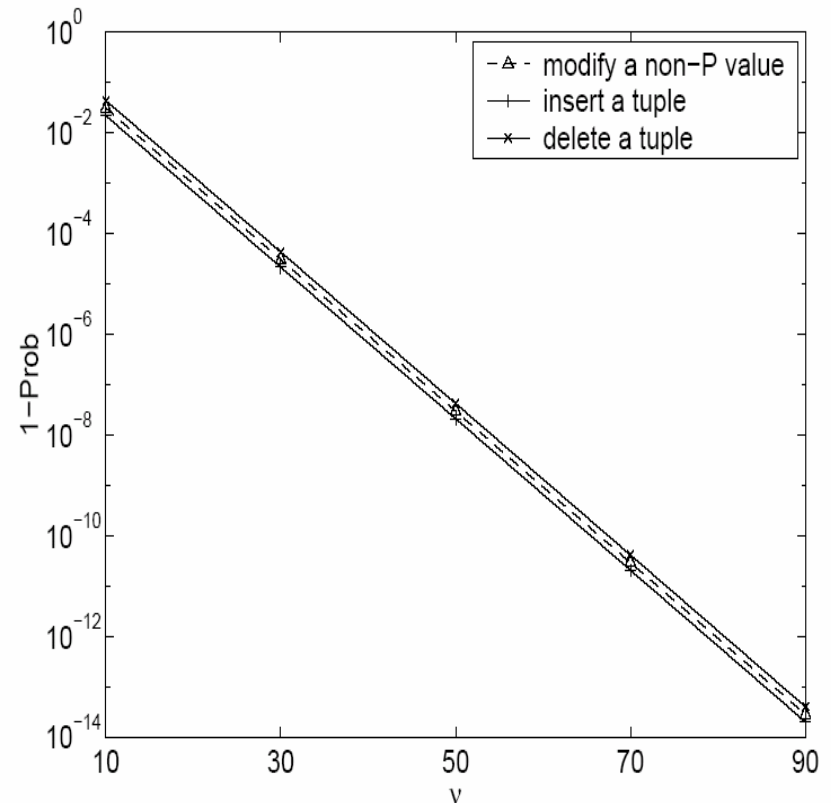


Figure 3: Error in detecting single value tamper

Modify multiple values

- Assume n tuples are modified
- The exact probability of all modifications being detected
 - Depends on how the n tuples distribute in different groups
 - Changing any number of tuples within a group has the same effect on watermarks
- For massive modification where $n > g$ and at least one tuple is modified in each group, then

$$Prob \simeq \left(1 - \frac{1}{2^{\frac{v}{2}}}\right)^g$$

- See paper for details and example

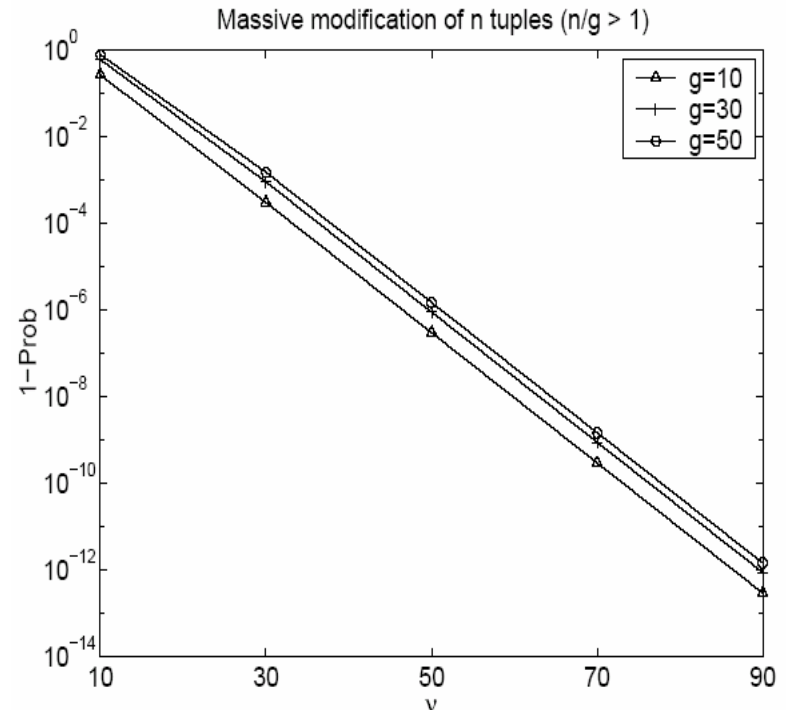


Figure 4: Error in detecting massive value modification

Insert multiple tuples

- Assume n tuples are inserted
- For massive insertion where $n > g$ and n/g tuples are inserted into each group on average, then

$$Prob \simeq \left(1 - \frac{1}{2^{\frac{\nu+n/g}{2}}}\right)^g$$

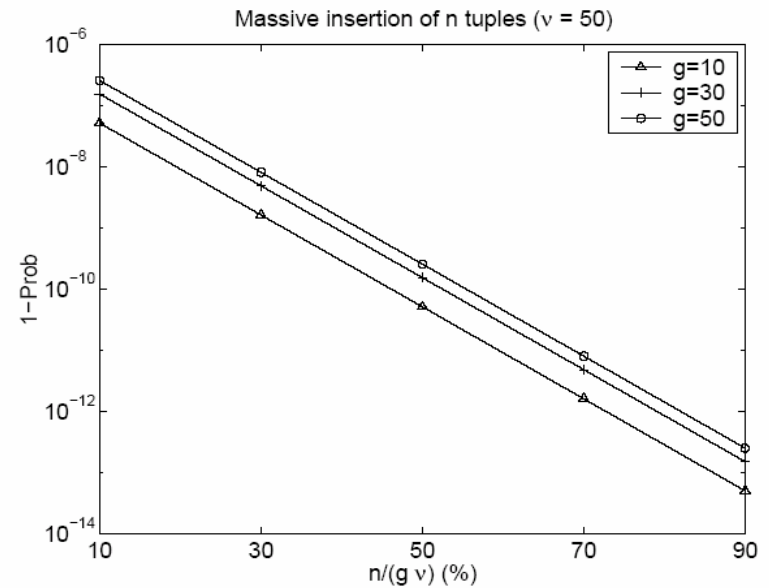
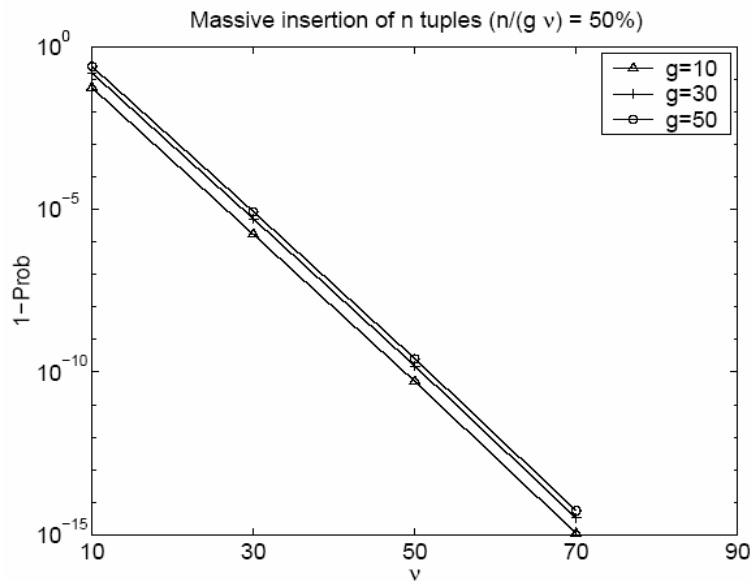


Figure 6: Error in detecting massive tuple insertion

Delete multiple tuples

- Assume n tuples are deleted
- For massive deletion where $n > a$ and n/g tuples are deleted from each page, then $Prob \simeq \left(1 - \frac{1}{2^{\frac{\nu - n/g}{2}}}\right)^g$, then

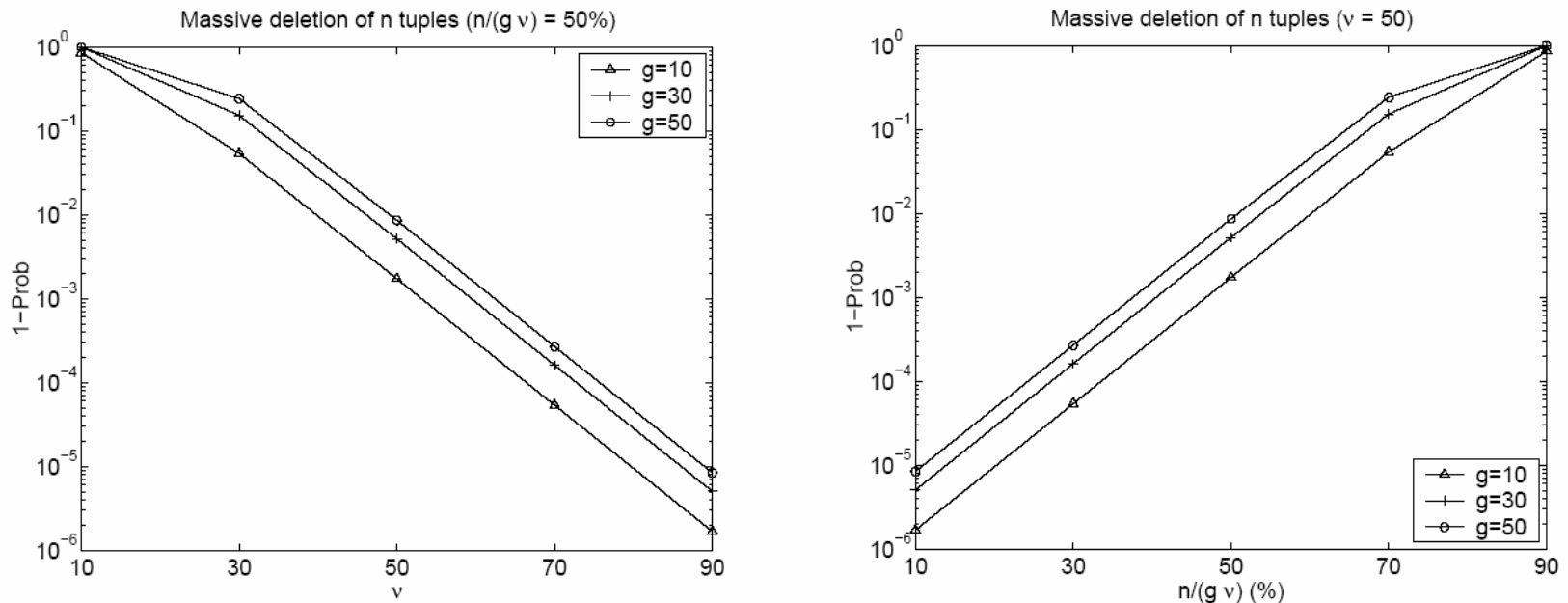


Figure 7: Error in detecting massive tuple deletion

Comparison

- With group size increasing, all error rates decrease
 - because the length of affected watermarks increases (linearly).
- With the same group size, error rates decrease in the order of
 - tuple deletion, modification, insertion
 - because the length of affected watermarks decreases, remains the same, and increases, respectively

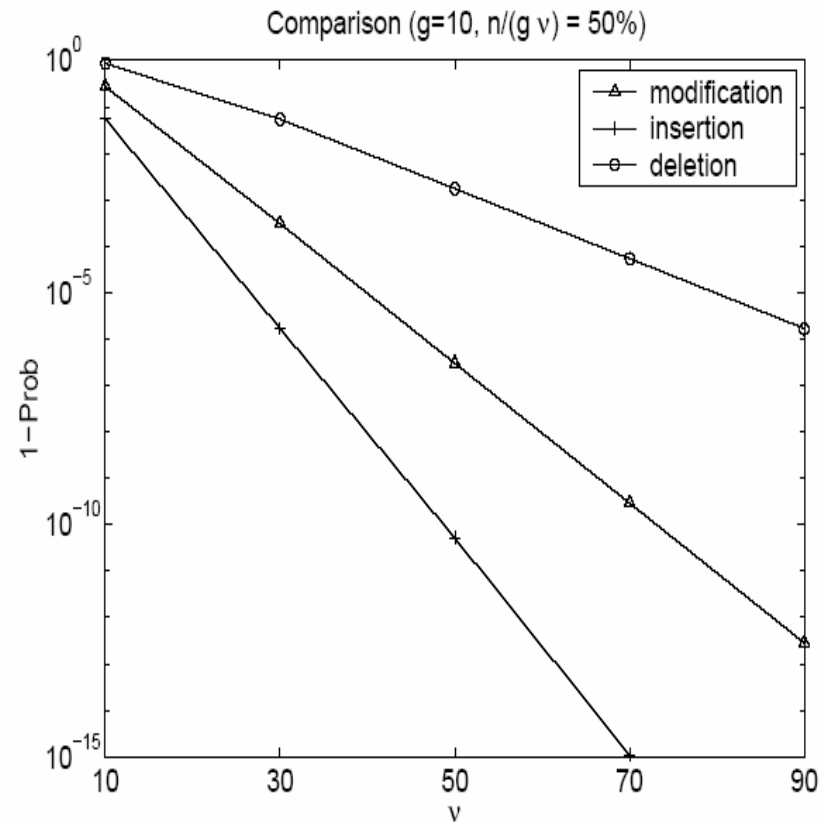


Figure 5: Error in detecting massive tamperers

Discussion on group size

- Trade-off between security and localization
 - The larger the group size, the larger the prob. of detecting modifications in watermark detection, and the more secure is the scheme
 - The larger the group size, the less precisely one can localize modifications

Conclusion

- An efficient authentication scheme for categorical data
- Strength
 - Detect modifications
 - Locate modifications
 - No space overhead
- Trade-off between security and localization

Thank you very much!

