Database Watermarking

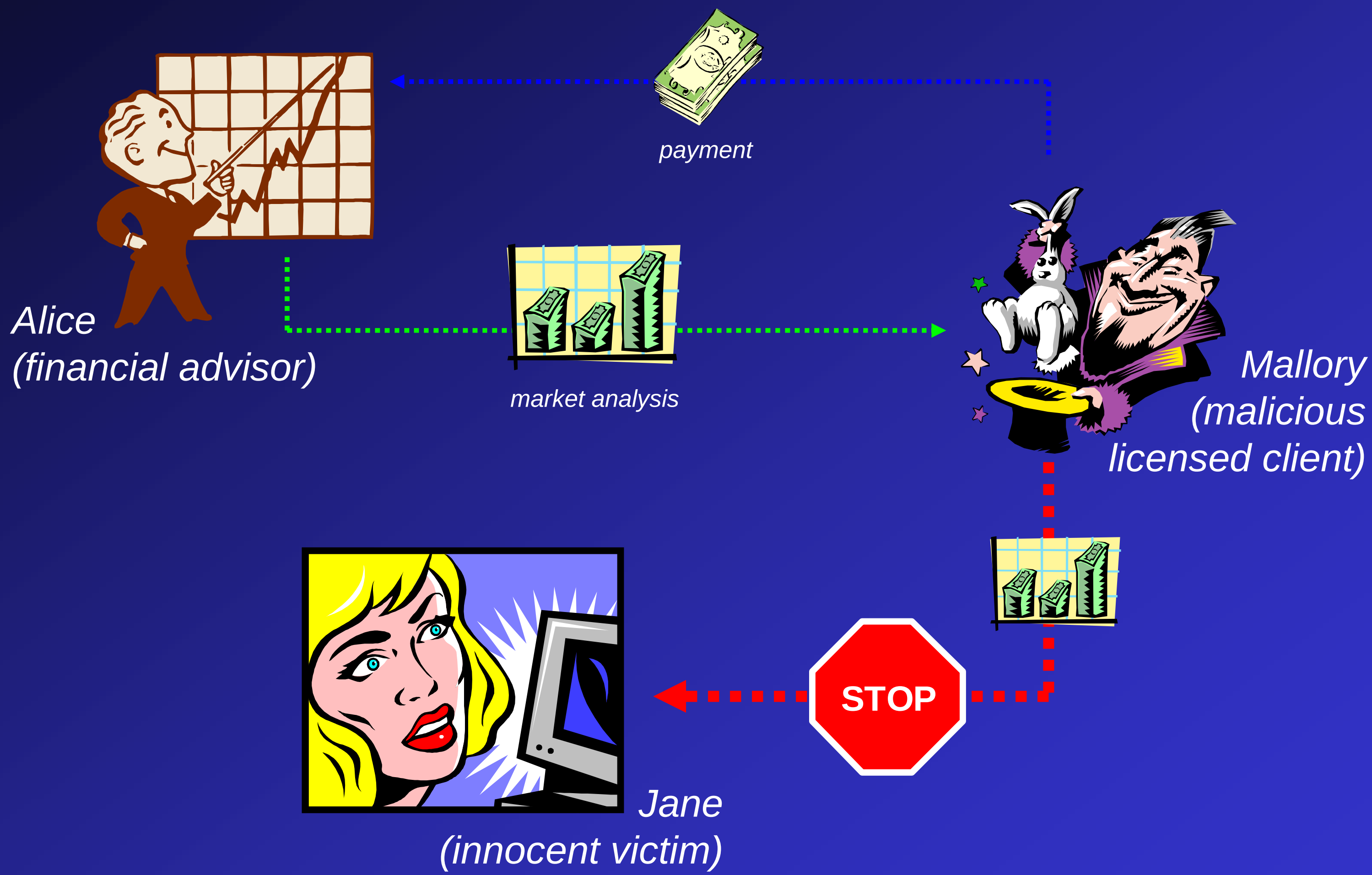
Radu Sion
(sion@cs.stonybrook.edu)
http://www.cs.stonybrook.edu/~sion

Computer Sciences
Stony Brook University
Stony Brook, NY 11794

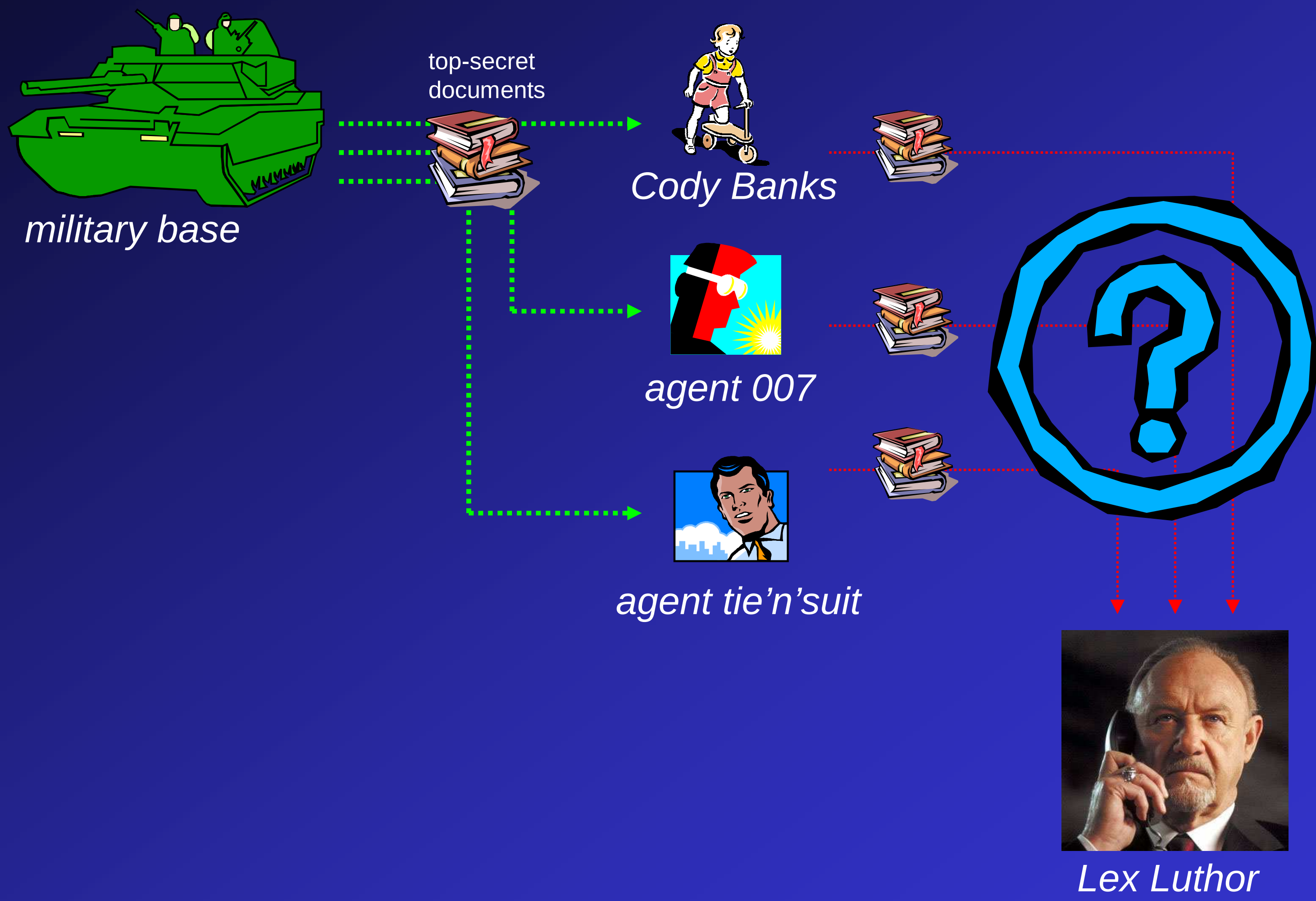
talk pointer

→ introduction
existing research: media
beyond media
numeric relational data
categorical data
sensor streams
limits of watermarking
the future

scenario


payment
Alice
(financial advisor)
market analysis
Mallory
(malicious
licensed client)
STOP
Jane
(innocent victim)

alternate scenario: traitor tracing



overview

Exponentially increasing amounts of *valuable* information we want to share.

Connected environments.

“Digital”: *zero-cost* verbatim copies.

→ Significant potential for misuse and illicit profit.

It becomes essential to have the integrated ability to *assert digital rights*.

→ “rights protection”

rights protection

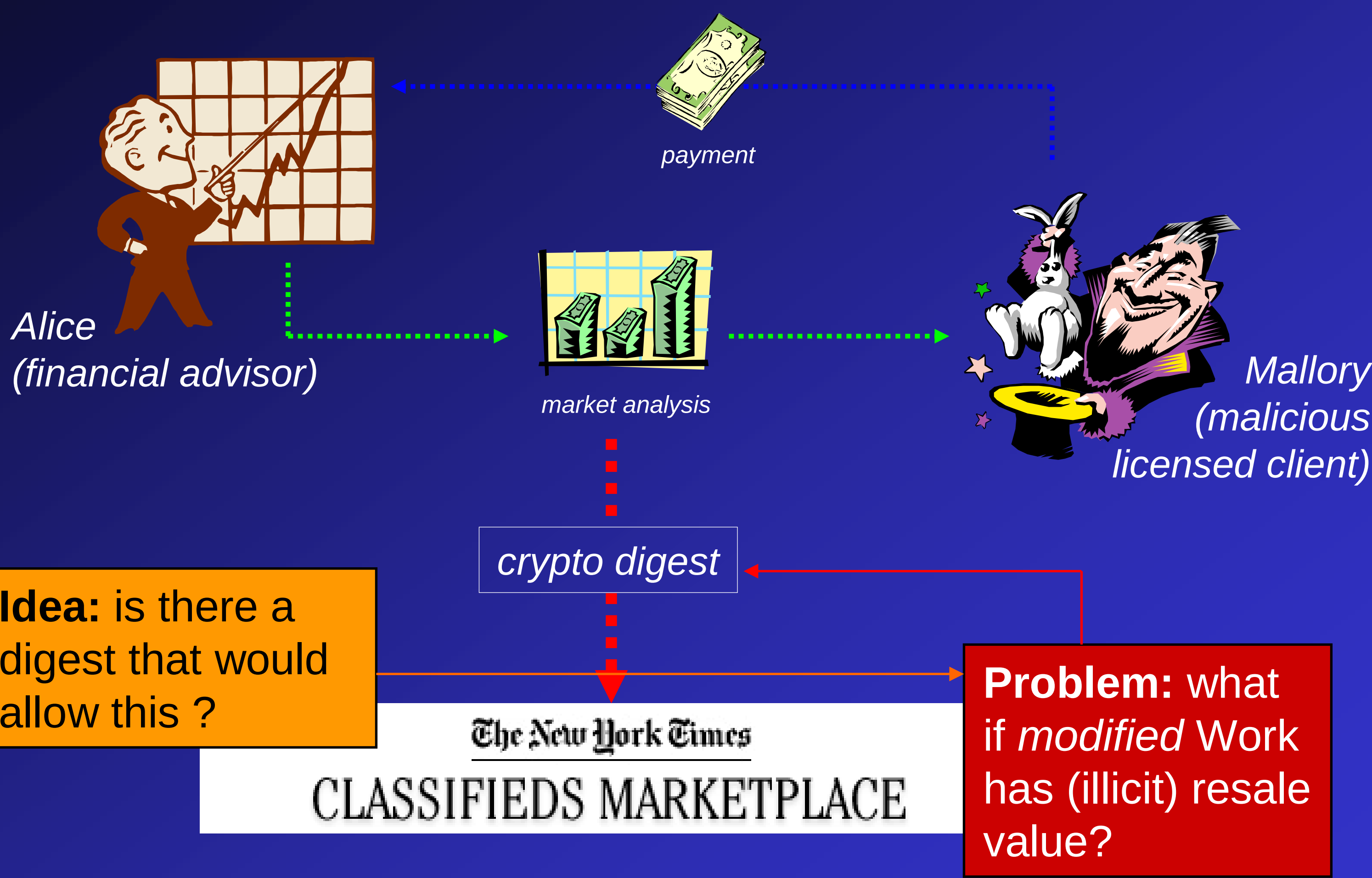
What ?

identity (e.g. rights holder) ↔ Work

How ?

→ legal means (e.g. severe penalties) +
→ technology (e.g. Watermarking)

but why not publish newspaper digest ?



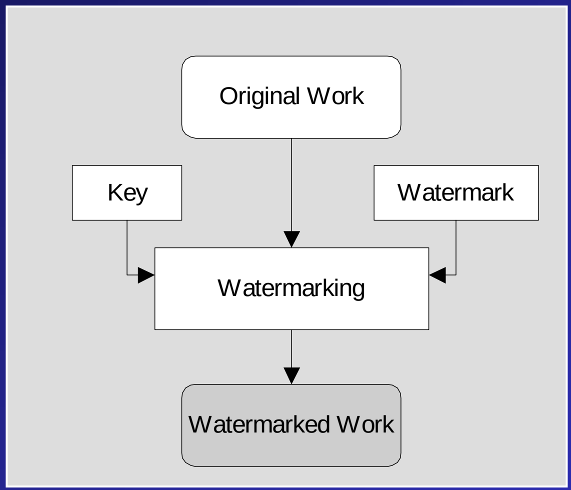
Idea: is there a digest that would allow this ?

Problem: what if *modified* Work has (illicit) resale value?

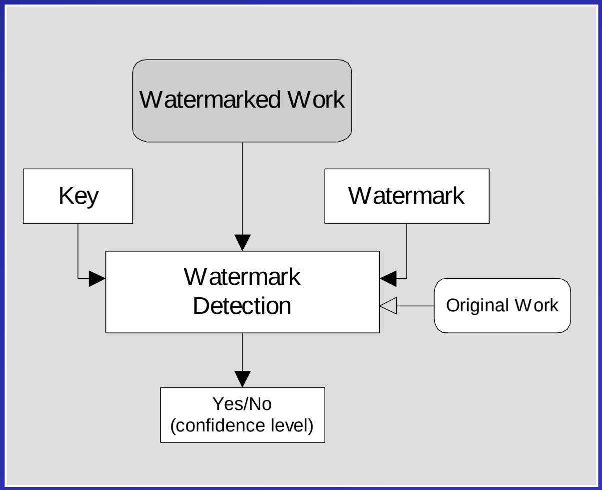
watermarking: rights protection tool

Watermarking induces a *convincing* and *relevant* (wrt. court-time proofs) property (**“rights signature”**) in a Work, through minor alterations.

- “convincing” ← very “rare” (false positives)
- “relevant” ← “© by radu”



watermark embedding



watermark detection

attacks: Mallory (evil entity)

Mallory wants to sell our data illicitly.

- → Detect and Remove (“subtractive”)
- → Perturb
- → Add new Watermark (“additive”)
- → Collude different watermarked copies

Watermarking is a game against Mallory !

talk pointer

- introduction
- → existing research: media
- beyond media
- numeric relational data
- categorical data
- sensor streams
- limits of watermarking
- the future

media

The encoding bandwidth stems from knowing the main digital Works consumer (*human*) and the associated limitations of perception.

→ metric of *distortion*
→ *allowable bounds*
→ "*resilience*"

talk pointer

- introduction
- existing research: media
- → beyond media
- numeric relational data
- categorical data
- sensor streams
- limits of watermarking
- the future

watermarking beyond media

In a general data domain (non-media) most existing (multimedia) techniques do not apply …

… because *distortion* metrics, *tolerable bounds*, and *resilience* often bear multiple semantics.

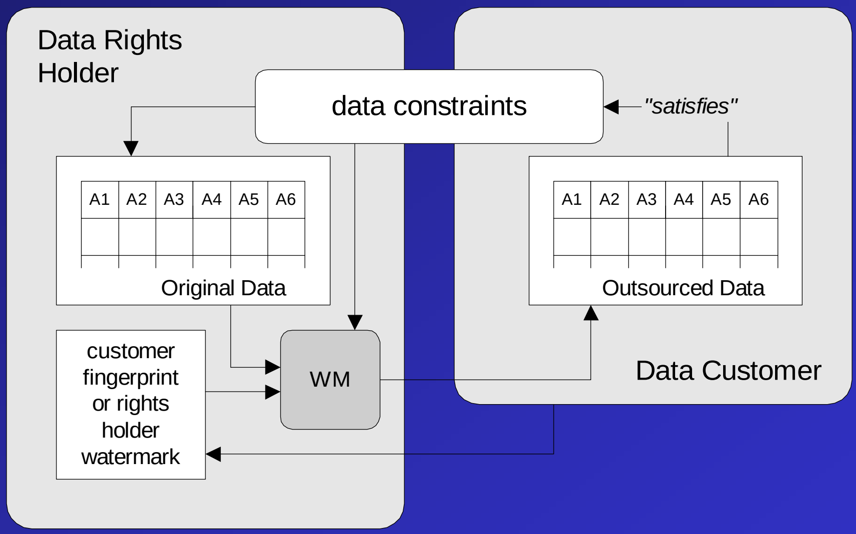
Q: How to preserve data value when there is always a semantic dimension that gets impacted even by minor changes …

… thus maybe we should instead focus on preserving application specific quality properties.

model: "consumer driven" watermarking

Data *consumer* requirements define distortion metrics and associated bounds. Encoding only guarantees them.

In other words: quality metrics should <u>not</u> be hard-coded but rather separated from the watermarking method.



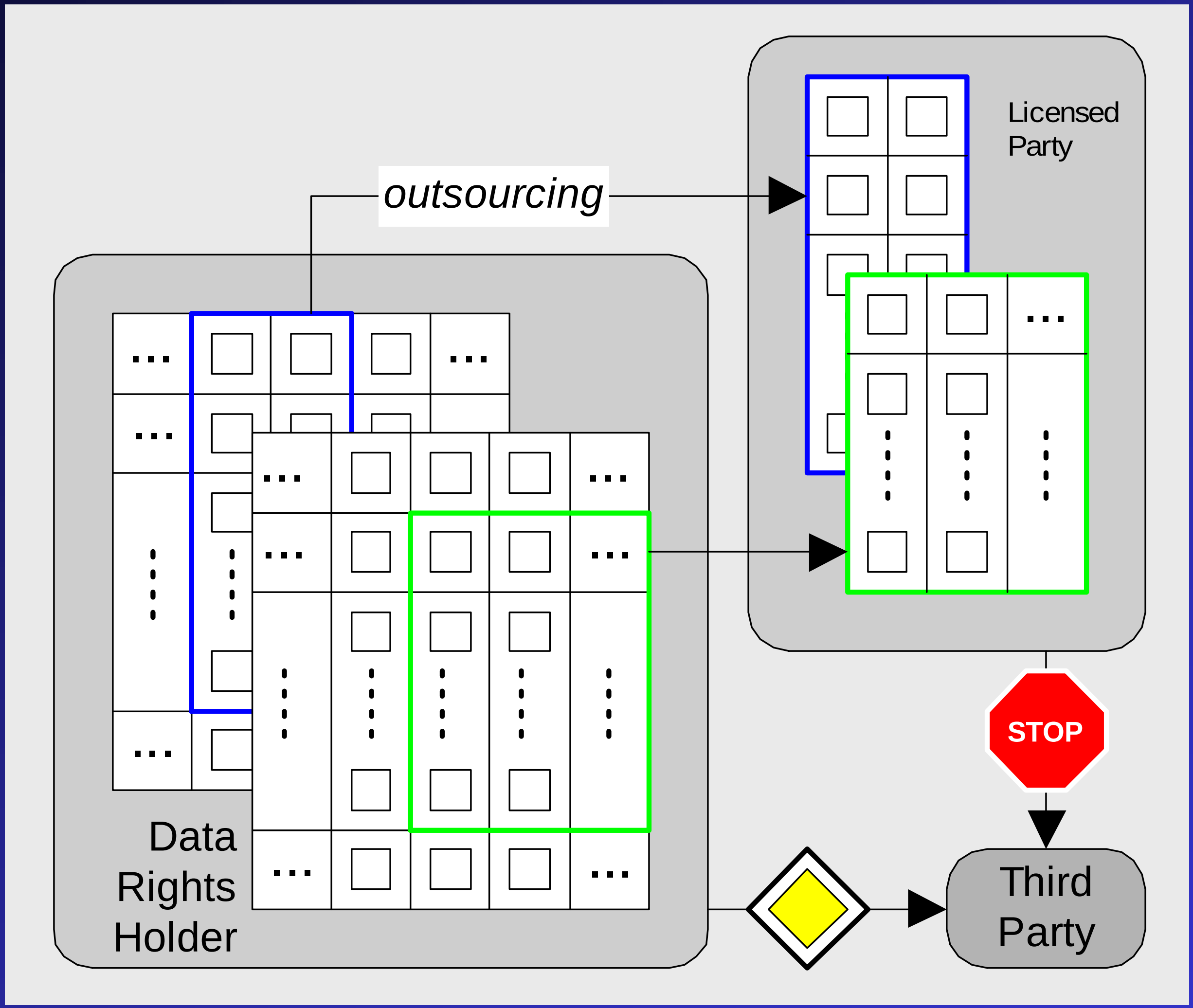
essential desiderata

Rights protection method should not interfere with intended data use.

talk pointer

introduction
existing research: media
beyond media
→ numeric relational data
categorical data
sensor streams
limits of watermarking
the future

relational data: scenario



relational data: problem

relational database **B**, set of quality constraints **C**

→ determine **B'** (a "watermarked" **B)**

B' satisfies **C** and features enough mark *resilience*.

→ what is "resilience" ?
→ what type of constraints ?
→ minimal context detection ("blind") ?

numeric data: initial ideas

First thoughts: randomly change bits in the data according to a certain criteria.

numeric data: Agrawal and Kiernan, 2002-03

// The private key $\mathcal{K}$ is known only to the owner of the database.
// The parameters γ, ν, and ξ are also private to the owner.

foreach tuple $r \in R$ **do**
  if ($\mathcal{F}(r.P)$ mod γ equals 0) **then** // mark this tuple
    attribute_index $i = \mathcal{F}(r.P)$ mod ν // mark attribute A_i
    bit_index $j = \mathcal{F}(r.P)$ mod ξ // mark j^{th} bit
    $r.A_i$ = mark($r.P, r.A_i, j$)

mark(primary_key pk, number v, bit_index j) **return** number

  first_hash = $\mathcal{H}(\mathcal{K} \circ pk)$

  if (first_hash is even) **then**
    set the j^{th} least significant bit of v to 0
  else
    set the j^{th} least significant bit of v to 1

  return v

watermark insertion

[1] Rakesh Agrawal, Peter J. Haas, and Jerry Kiernan. Watermarking relational data: framework, algorithms and analysis. The VLDB Journal, 12(2):157-169, 2003.

numeric data: Agrawal and Kiernan, 2002-03

watermark insertion

// The private key $\mathcal{K}$ is known only to the owner of the database.
// The parameters γ, ν, and ξ are also private to the owner.

1) **foreach** tuple $r \in \mathcal{R}$ **do**
2) **if** ($\mathcal{F}(r.P)$ mod γ equals 0) **then** // mark this tuple
3) attribute_index $i = \mathcal{F}(r.P)$ mod ν // mark attribute A_i
4) bit_index $j = \mathcal{F}(r.P)$ mod ξ // mark j^{th} bit
5) $r.A_i$ = mark($r.P, r.A_i, j$)

6) mark(primary_key pk, number v, bit_index j) **return** number

7) first_hash = $\mathcal{H}(\mathcal{K} \circ pk)$

8) **if** (first_hash is even) **then**
9) set the j^{th} least significant bit of v to 0
10) **else**
11) set the j^{th} least significant bit of v to 1

12) **return** v

η	Number of tuples in the relation
ν	Number of attributes in the relation available for marking
ξ	Number of least significant bits available for marking in an attribute
$1/\gamma$	Fraction of tuples marked
ω	Number of tuples marked
α	Significance level of the test for detecting a watermark
τ	Minimum number of correctly marked tuples needed for detection

[1] Rakesh Agrawal, Peter J. Haas, and Jerry Kiernan. Watermarking relational data: framework, algorithms and analysis. The VLDB Journal, 12(2):157-169, 2003.

numeric data: Agrawal and Kiernan, 2002-03

watermark detection

// $\mathcal{K}$, γ, ν, and ξ have the same values used for watermark insertion.
// α is the test significance level that the detector preselects.

1) totalcount = matchcount = 0

2) **foreach** tuple $s \in \mathcal{S}$ **do**
3) **if** ($\mathcal{F}(s.P)$ mod γ equals 0) **then** // this tuple was marked
4) attribute_index $i = \mathcal{F}(s.P)$ mod ν // attribute A_i was marked
5) bit_index $j = \mathcal{F}(s.P)$ mod ξ // j^{th} bit was marked
6) totalcount = totalcount + 1
7) matchcount = matchcount + match($s.P$, $s.A_i$, j)

8) τ = threshold(totalcount, α) // see Section 4.2
9) **if** (matchcount $\geq \tau$) **then** *suspect piracy*

10) match(primary_key pk, number v, bit_index j) **return** int

11) first_hash = $\mathcal{H}(\mathcal{K} \circ pk)$

12) **if** (first_hash is even) **then**
13) return 1 if the j^{th} least significant bit of v is 0 else return 0
14) **else**
15) return 1 if the j^{th} least significant bit of v is 1 else return 0

η	Number of tuples in the relation
ν	Number of attributes in the relation available for marking
ξ	Number of least significant bits available for marking in an attribute
$1/\gamma$	Fraction of tuples marked
ω	Number of tuples marked
α	Significance level of the test for detecting a watermark
τ	Minimum number of correctly marked tuples needed for detection

numeric data: virtual primary key

- If primary key does not exist or could be changed in attacks, then for each tuple
  - partition each attribute Ai into MSBs Mi and LSBs (least g significant bits) Li
  - virtual primary key = concatenation of two (or more) hash values in set {H(K,Mi): i=0,…r-1} that are closest to zero

• Dynamic: for different tuples, different attributes may be selected (based on secret key) to form virtual primary key

• Content-based: it depends on hash values of MSBs rather than order of attributes

[4] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. Constructing a virtual primary key for fingerprinting relational data. In DRM '03: Proceedings of the 2003 ACM workshop on Digital rights manage ment, pages 133-141, New York, NY, USA, 2003. ACM Press.

numeric data: challenges

Challenges:

→ sensitive data (destroys ulterior data uses)
→ natural numeric transformations

It is necessary:

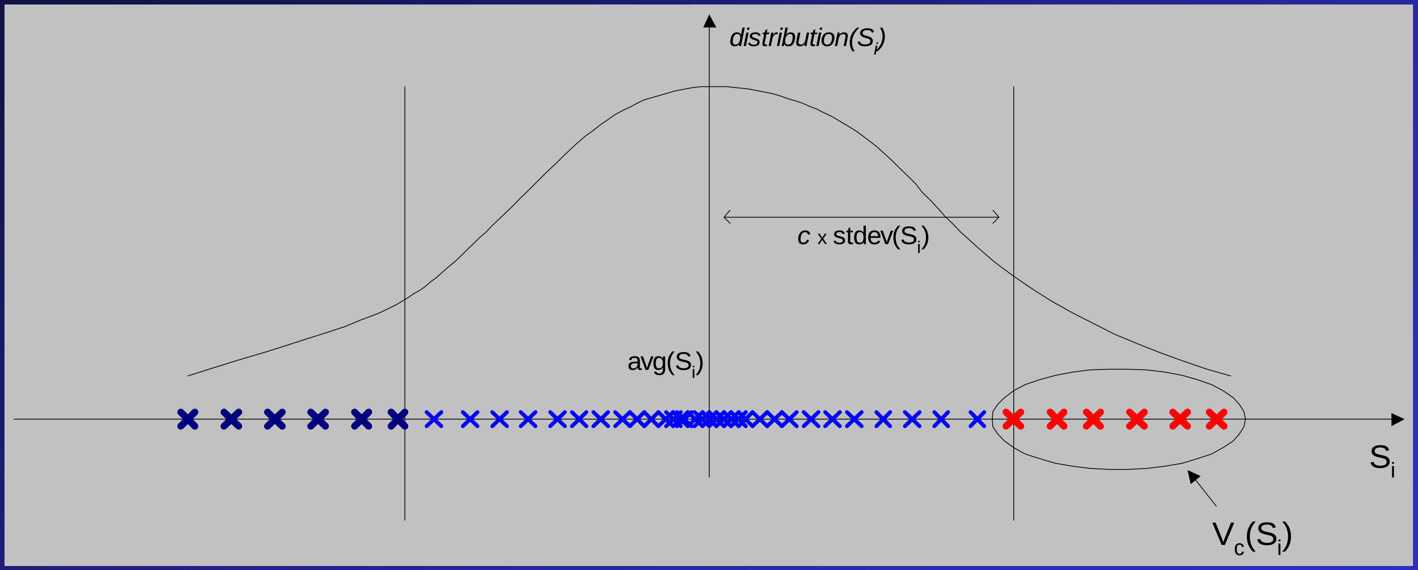
→ to handle a set of desired quality metrics +
→ survive attacks (e.g. segmentation, alterations)

numeric data: key insight

Encode information in *global numeric properties* of *secret subsets* of the data while continuously evaluating data quality **C** (backtrack if necessary).

→ *Which* properties ?
→ *How* do we use them ?

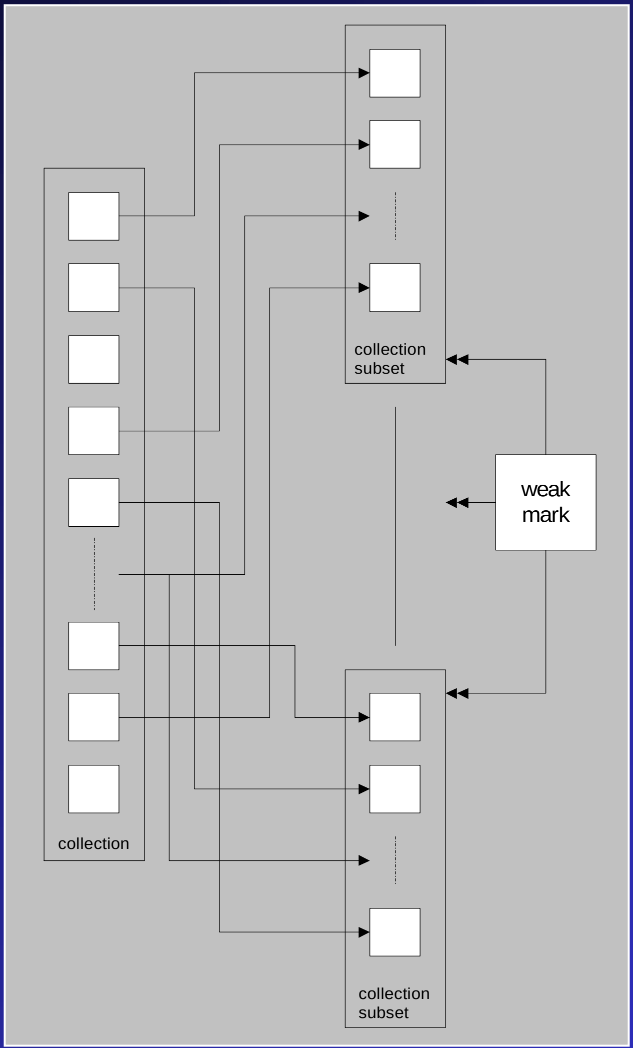
numeric data: weak mark

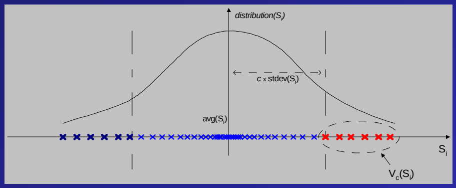


handles:

→ data loss
→ linear changes
→ random alterations

numeric data: amplification

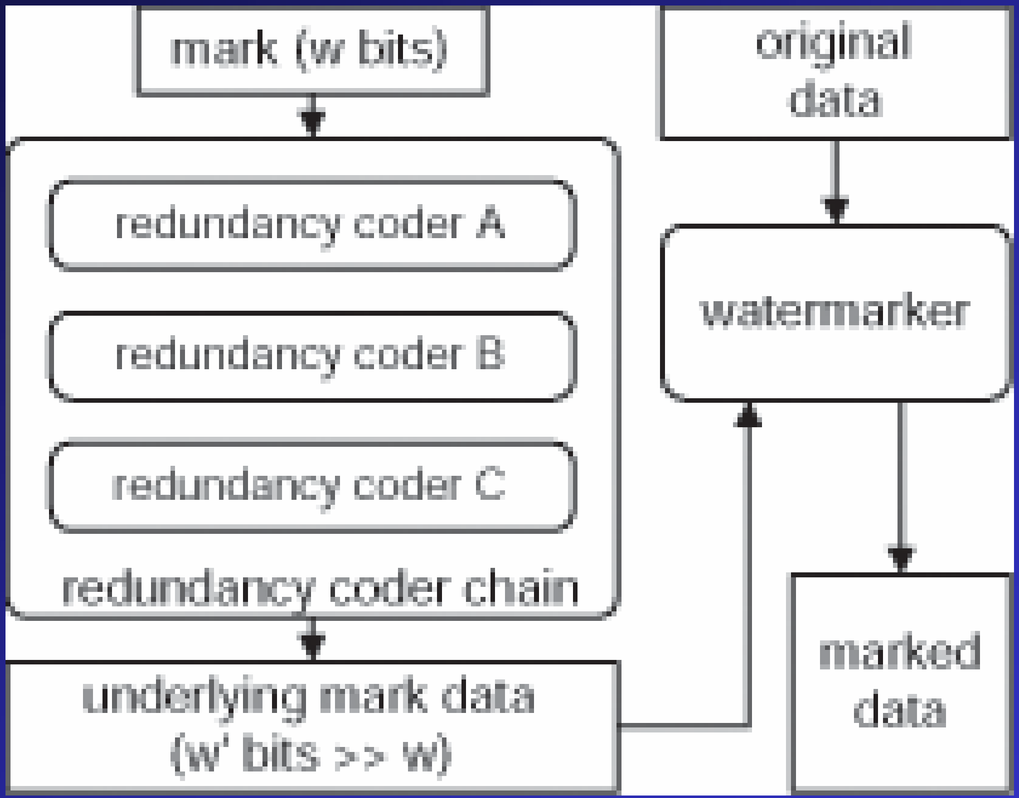




“amplification”

By applying a (weak) mark on ***secret*** subsets of the original data set, it is effectively *amplified*.

numeric data: error correction



numeric data: detection process

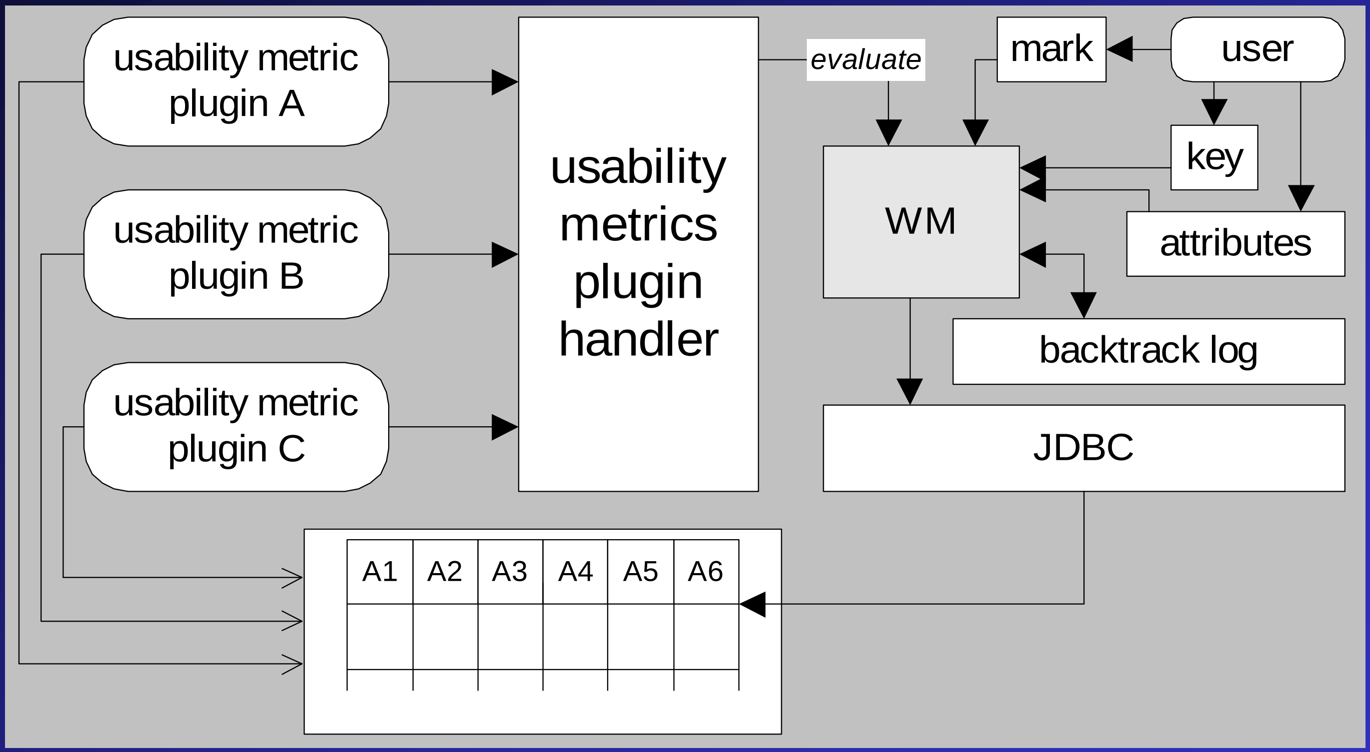
- reconstruct subsets
- detect invalid subset encoding(s)
- detect valid bits
- construct error correction map
- apply error correction chain
- reconstruct watermark

allowable semantic constraints

Idea: data *consumer* requirements define the data quality metrics and associated allowed bounds; encoding process guarantees those bounds.

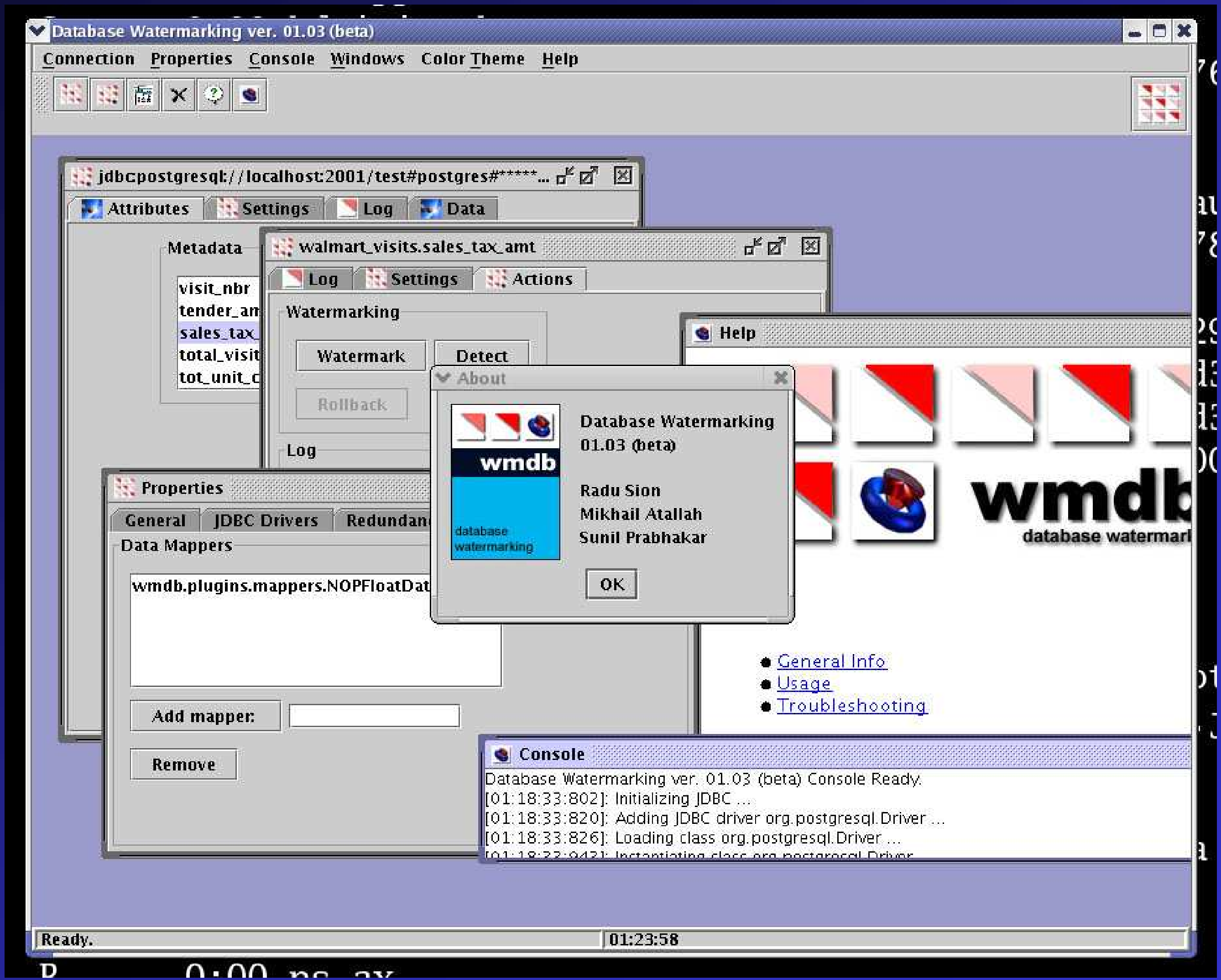
In other words: data quality metrics are ***not*** to be hard-coded; data use semantics are to be separated from the watermarking method.

wmdb.*: system architecture



- → multi-threaded
- → Java, tested with pgres/file-io/Oracle 9
- → different JDBC drivers for different connections
- → multiple databases at the same time, parallel watermarking

wmdb.*: runtime snapshot

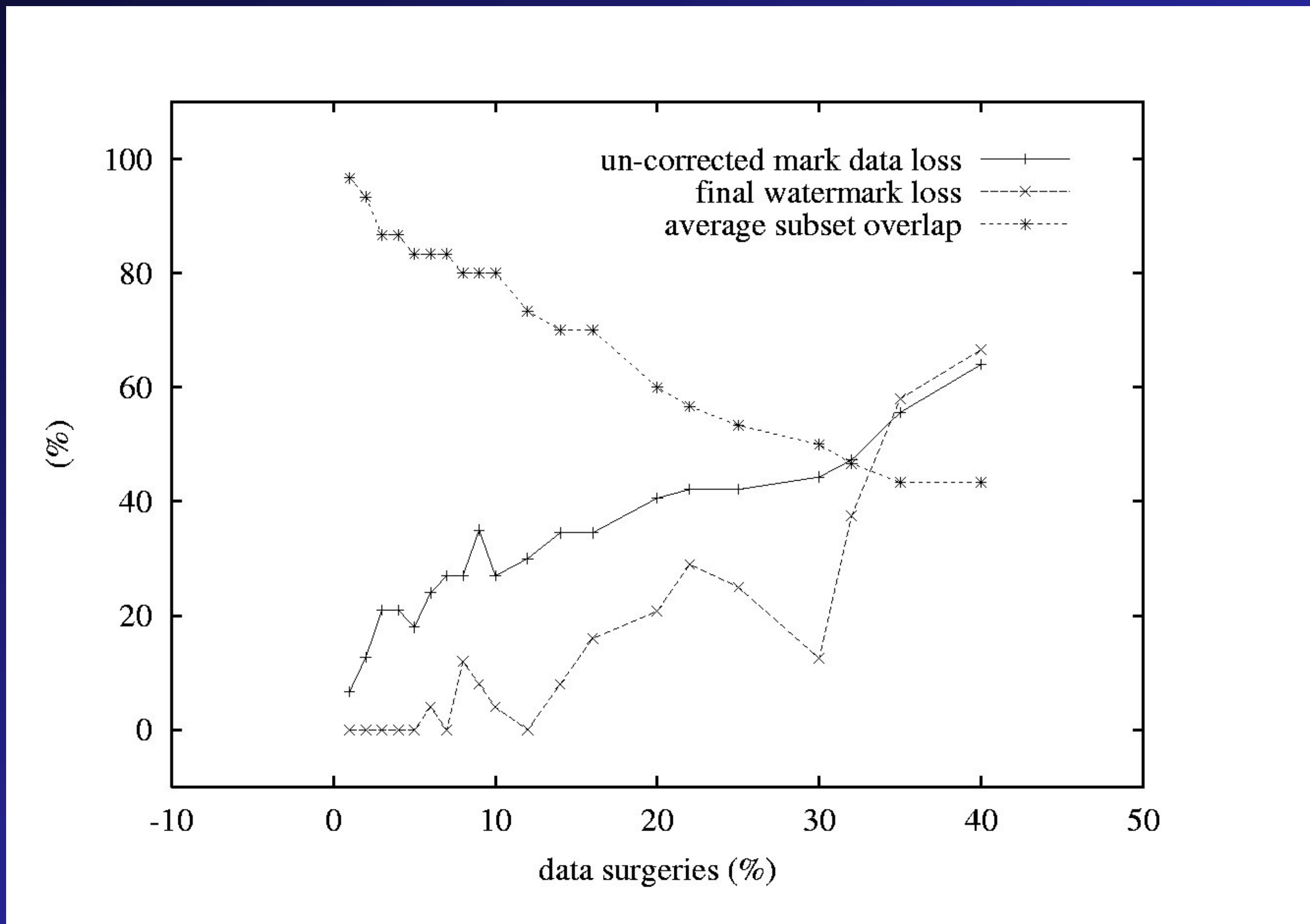


wmdb.*: experiments overview

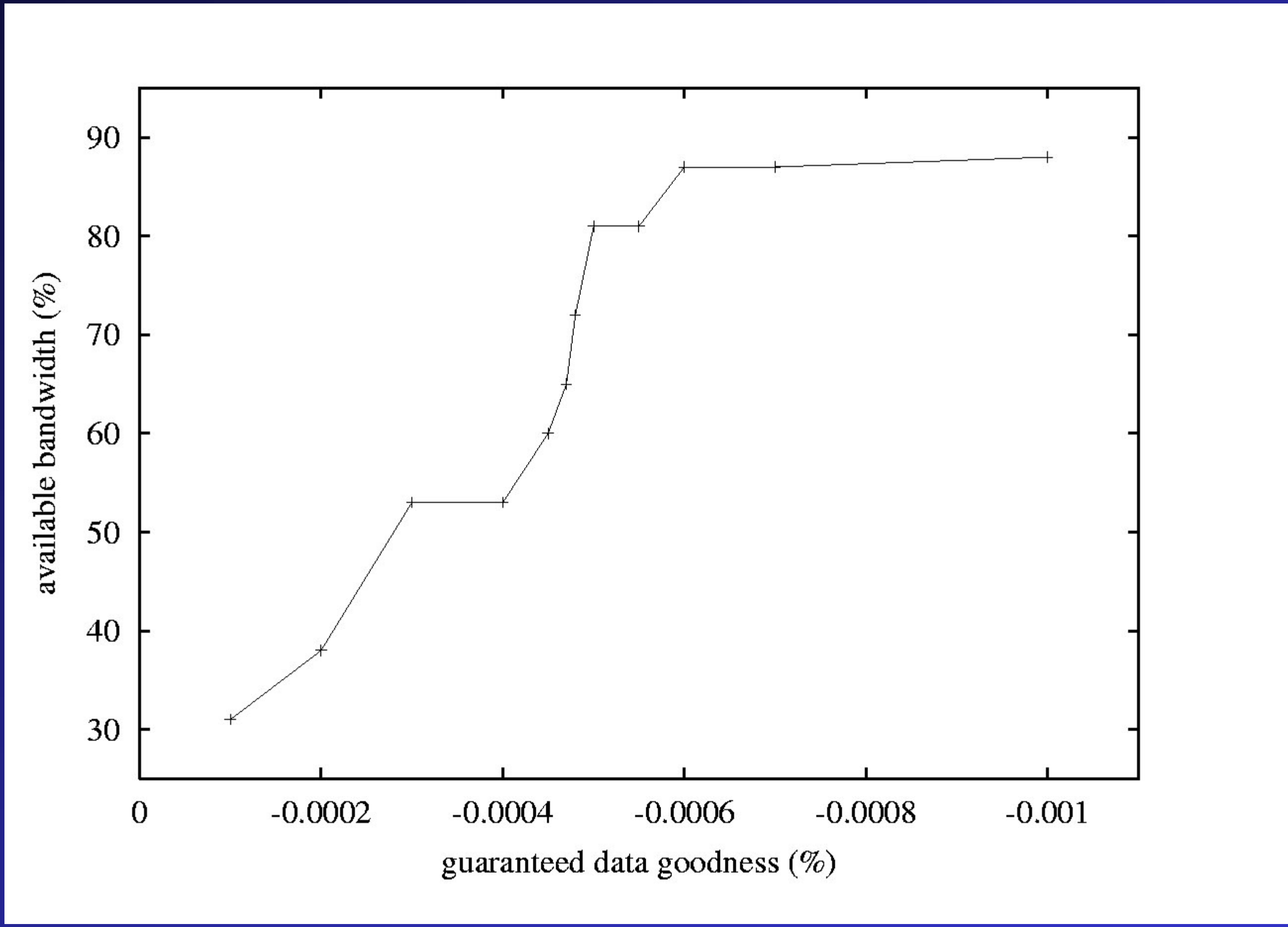
- resilience
  - data loss (subset selection)
  - absolute data changes
  - epsilon attacks
- semantics and data quality
  - classification preservation
  - arbitrary query result
- performance
  - running times

notes: PC, 256-1024MB RAM, Java, remote JDBC (SQL-92, e.g. Oracle, Postgres, even files) , simultaneous multiple database marking, Sales Data from Wal-Mart etc.

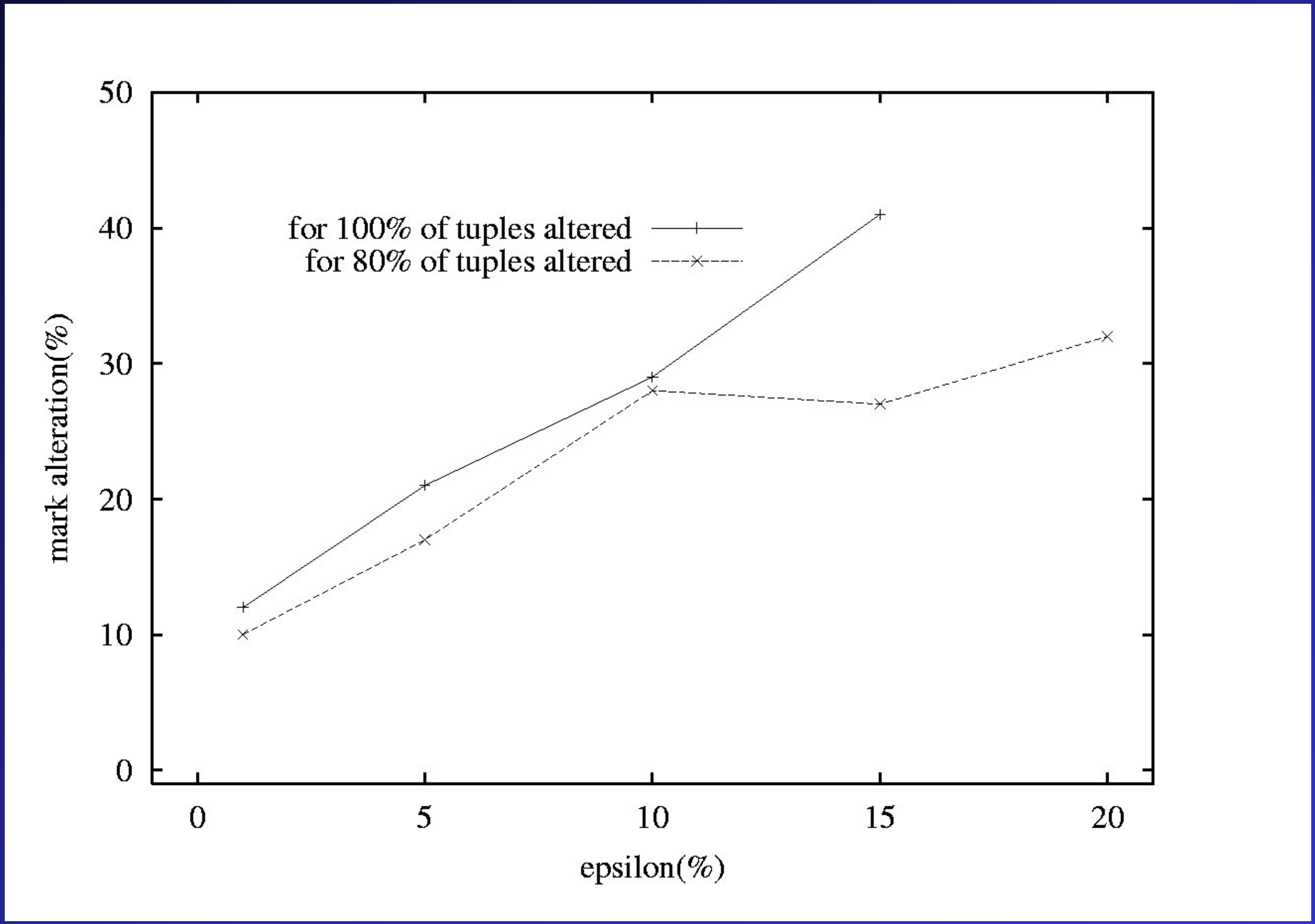
experiments: data loss resilience



experiments: absolute change

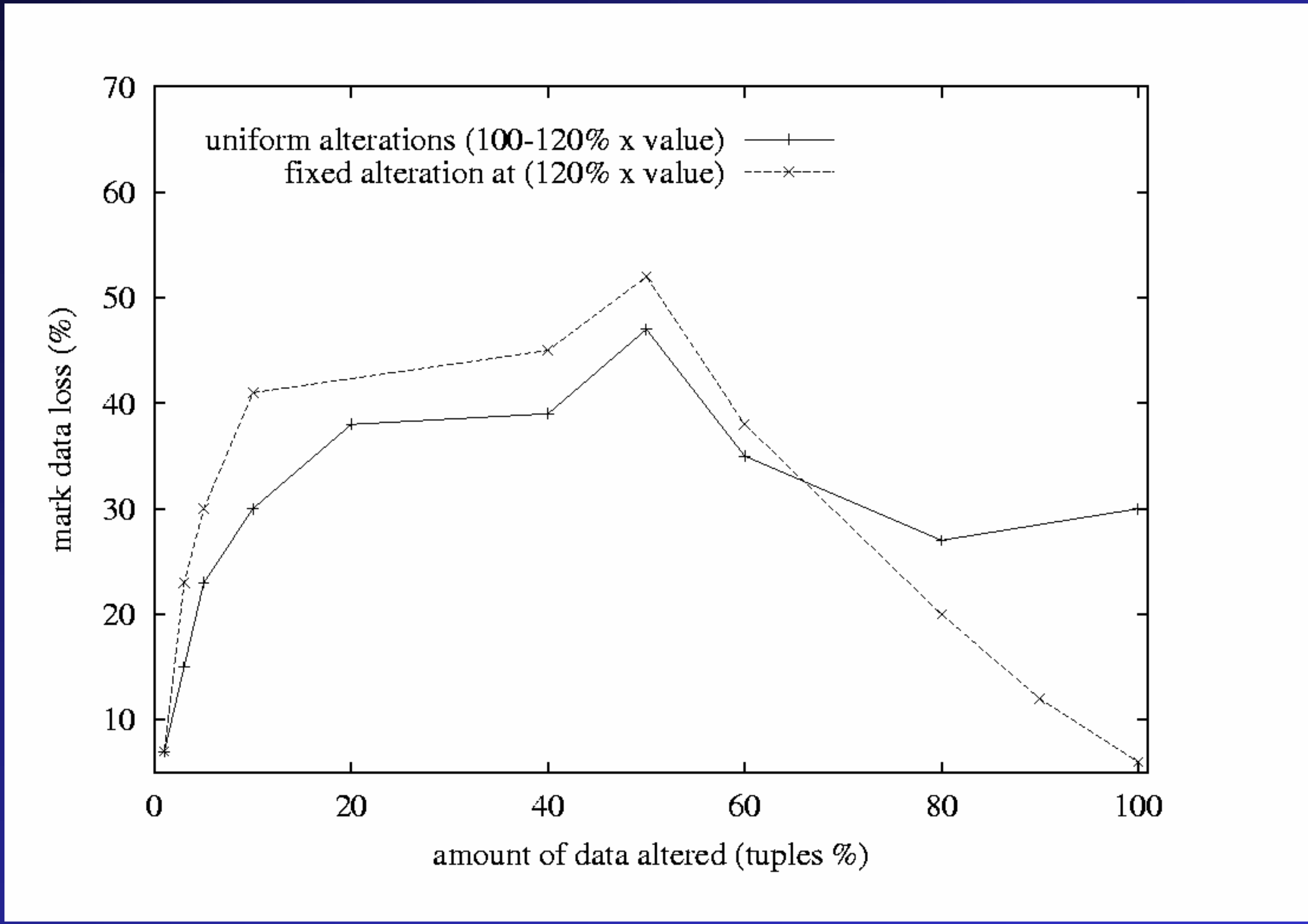


experiments: arbitrary data change



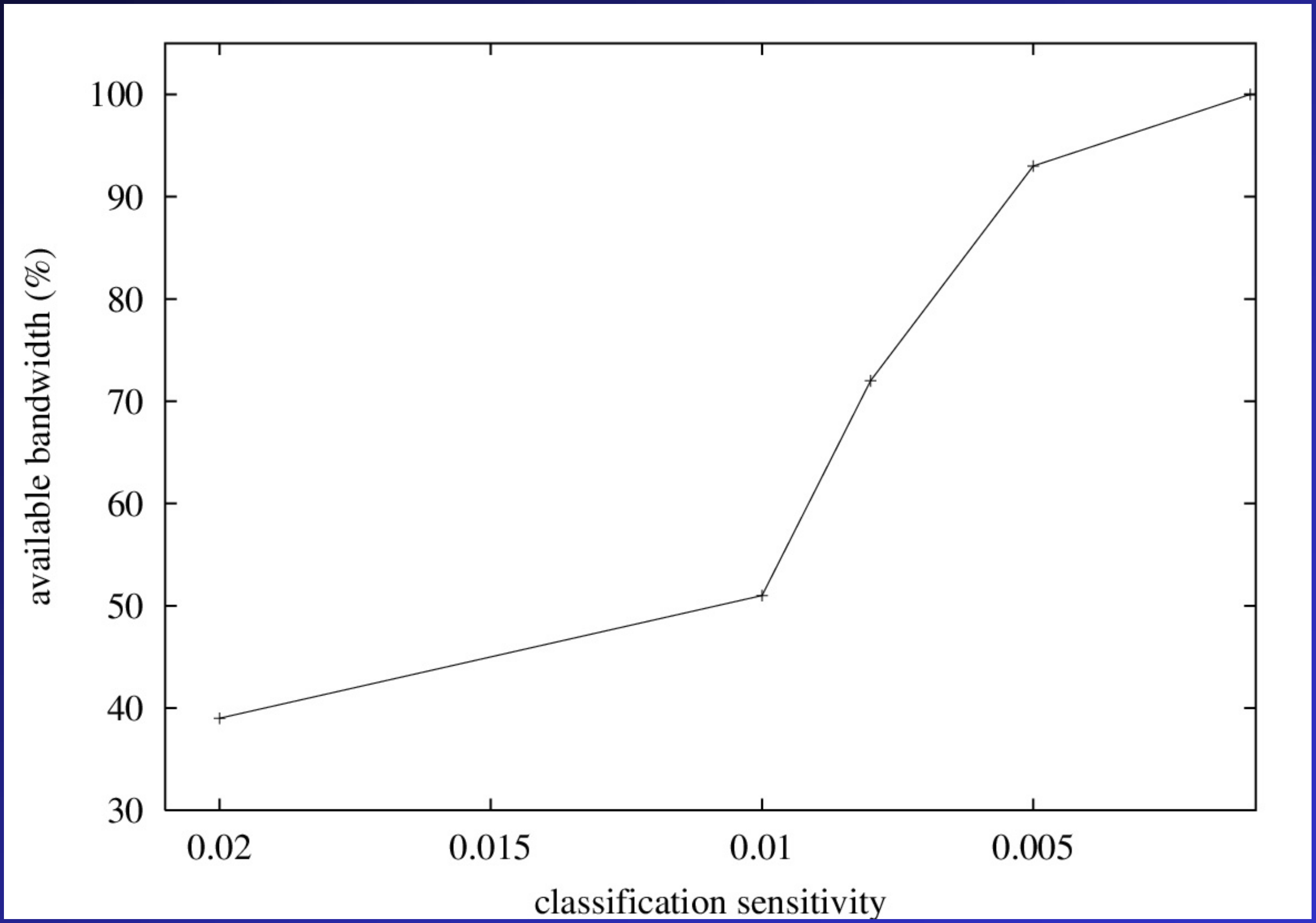
note: *average zero epsilon-attack*

experiments: arbitrary data change (2)



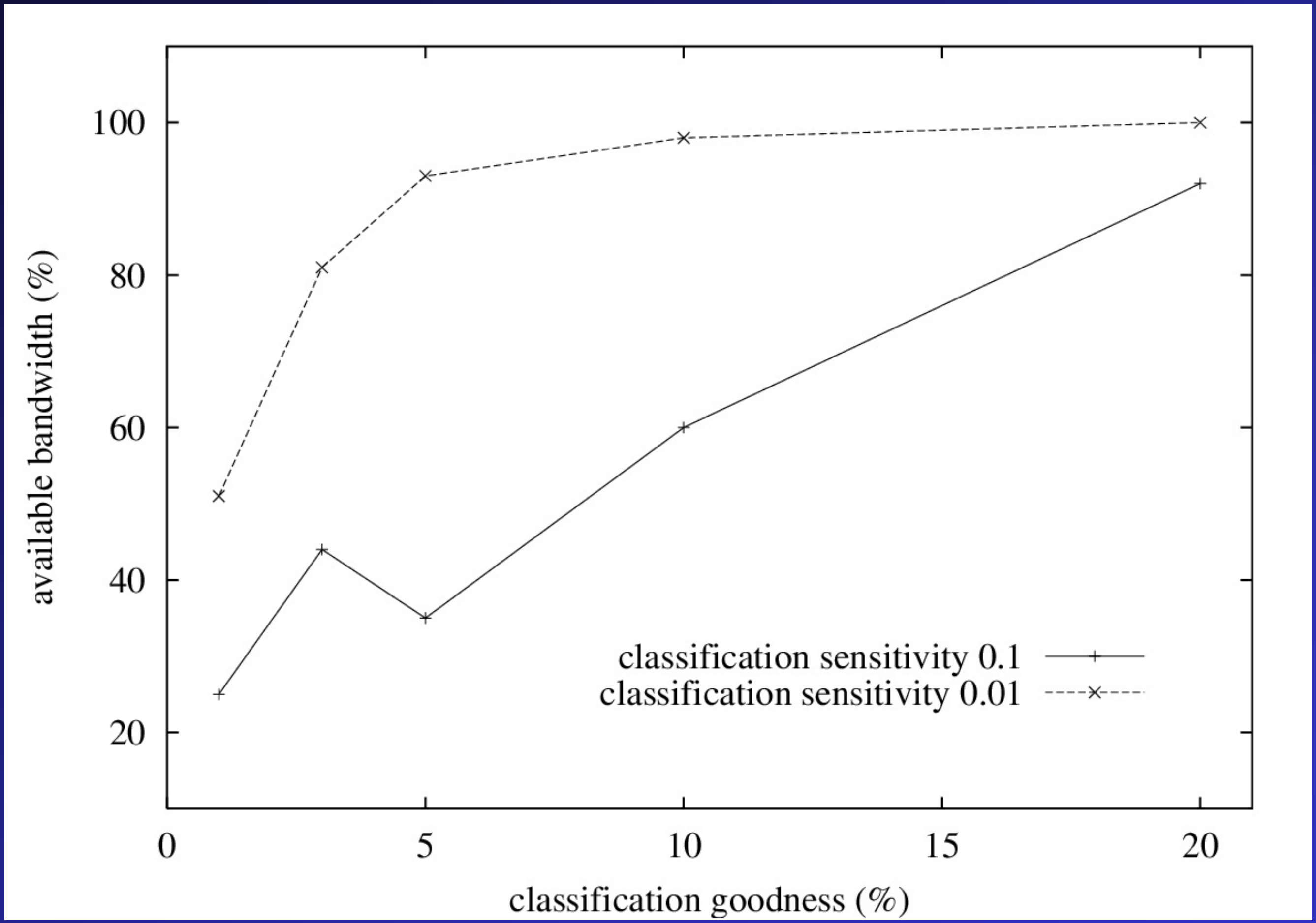
note: *average non-zero epsilon-attack, nice resilience, no knowledge of nature of transform.*

experiments: classification preservation



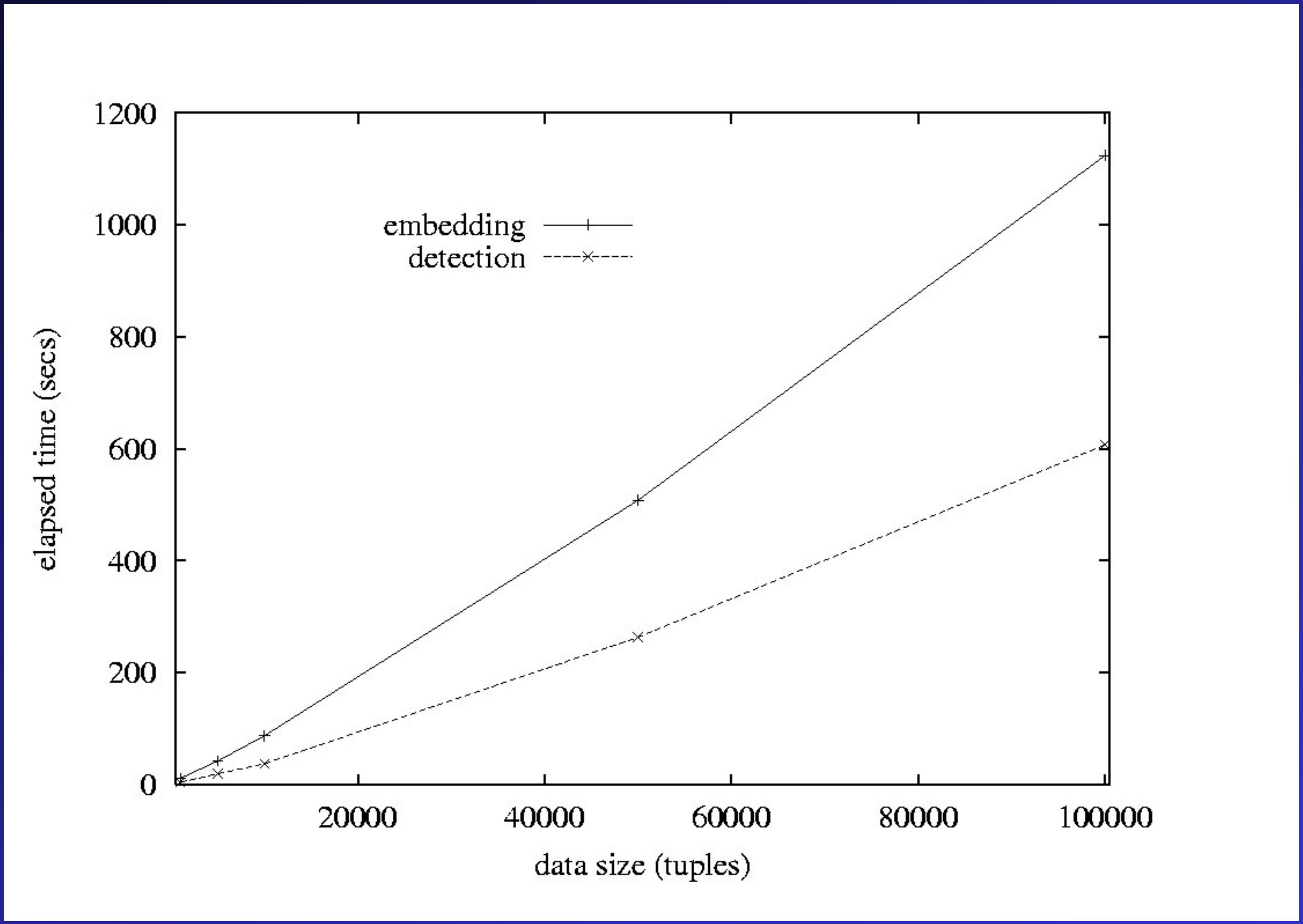
note: *as guaranteed classification sensitivity decreases, available bandwidth increases*

experiments: classification preservation (2)



note: *bandwidth increases as guaranteed classification goodness relaxes*

experiments: performance



note: *includes local network costs, DBMS costs*

talk pointer

introduction
existing research: media
beyond media
numeric relational data
→ categorical data
sensor streams
limits of watermarking
the future

categorical data: challenges

Because there are no epsilon-changes, earlier approaches (numeric) do not work.

- What is our embedding channel ?

  $\rightarrow$ statistical bias in inter-attribute association

- Any alteration is discrete, possibly significant

  $\rightarrow$ we would like to minimize the “number” (+maximize impact) of required alterations

embedding channel

K	A
0	a_3
1	a_7
2	a_9
3	a_2
…	
i	a_6
…	
n-1	a_8
n	a_7

$A \in \{a_1, \ldots, a_{n_A}\}$ (e.g. {"Chicago","Bucharest" … "Amsterdam"})

Because there are no epsilon-changes, earlier approaches (numeric) do not work.

$\Rightarrow$ we need a *different embedding channel* !

$\rightarrow$ inter-attribute association

Any alteration is discrete, possibly significant.

$\Rightarrow$ we would like to *minimize the "number" (+maximize impact)* of required alterations

single-key bias

Question: is the data watermarked ?

problem: is this "relevant"
(i.

solution: key K is "© by radu" (not viable)

solution: *multi-bit* watermark stating "© by radu"

select "fit" tuples

K	A
2	a_9
3	a_2
…	
i	$a_{f(i)}$
…	
n-1	a_8
n	a_7

$$f(i)=msb(H(i,k),\log_2(n_A))$$

How: slightly alter **A**, modulating some of its ("fit") values according to a keyed one-way hash of **K**

single-key, single-bit bias

Question: is the data watermarked ? if yes then what is the *one bit* watermark ?

K	A
0	a_3
1	a_7
2	a_9
3	a_2
...	
i	$a_{f(i)}$
...	
n-1	a_8
n	a_7

$$f(i)=msb(H(i \oplus w,k),\log_2(n_A))$$

detection time: "counting" bias

$msb(f(i),\log_2(n_A))==H(i \oplus \text{"true"},k) \rightarrow bias_{true}{+}{+}$

$msb(f(i),\log_2(n_A))==H(i \oplus \text{"false"},k) \rightarrow bias_{false}{+}{+}$

$\Rightarrow$ watermark: $func(bias_{true} - bias_{false})$

How: slightly alter **A**, modulating some of its ("fit") values according to a one-way keyed hash of **K** *and the value of the watermark bit **w***.

single-key, multi-bit bias

Question: is the data watermarked ? if ...hen what is the watermark *string* ?

problem: some watermark
b...
d...

solution: use separate keys $\mathbf{k_1}$ and $\mathbf{k_2}$ to de-correlate.

2	a_9
3	a_2
...	
i	$a_{f(i)}$
...	
n-1	a_8
n	a_7

		[3]	[4]	...	[g(i)]	...	[m-1]	[m]
$bias_{true}[1]$	...							
$bias_{false}[1]$	...							

$g(i)=msb(H(i,\mathbf{k_2}),\log_2(m))$

$f(i)=msb(H(i\oplus w[g(i)],\mathbf{k_1}),\log_2(n_A))$

How: slightly alter **A**, modulating some of its ("fit") values according to a one-way hash of **K** and a *spread of the values of the watermark* ***w****.*

multi-key, multi-bit bias

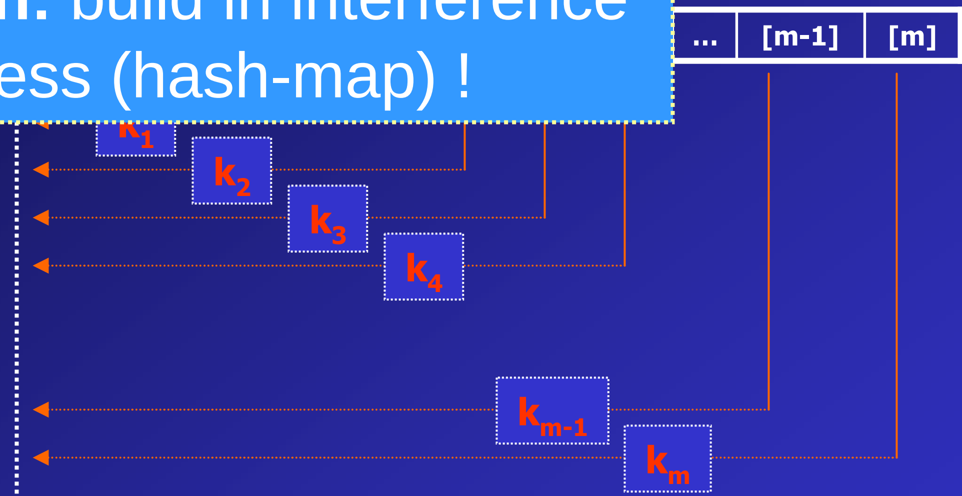
Question: is the data watermarked ? if … is the watermark *string* ?

drawback: multi-mark inference

mu…

solution: build in interference awareness (hash-map) !

0	a_3
1	a_7
2	a_9
3	a_2
…	
i	a_j
…	
n-1	a_8
n	a_7



global key → $\mathbf{k=(k_1,k_2,\ldots,k_m)}$

one-bit watermarking algorithm

How: embed **m** one-bit watermarks into data using separate dedicated keys.

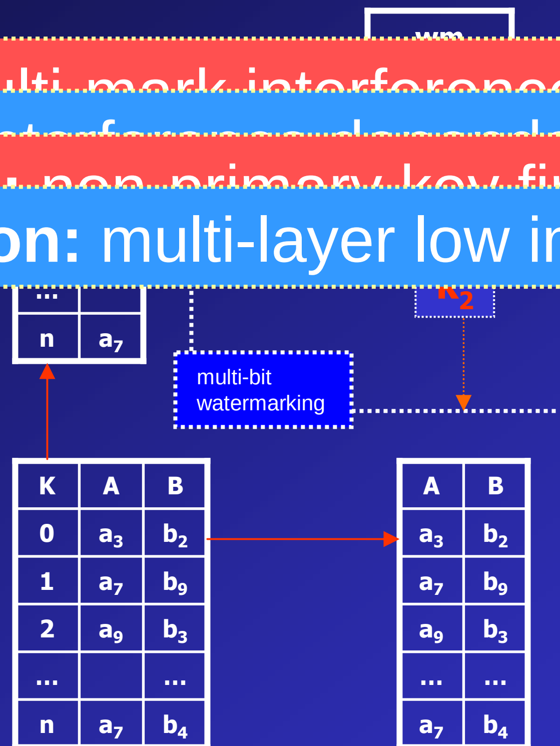
vertical partitioning: multi-attribute encoding

Question: How to survive vertical partitioning ?

problem: multi-mark interference

problem: non-primary key first argument

solution: multi-layer low impact watermarks



K	A	B
0	a_3	b_2
1	a_7	b_9
2	a_9	b_3
…		…
n	a_7	b_4

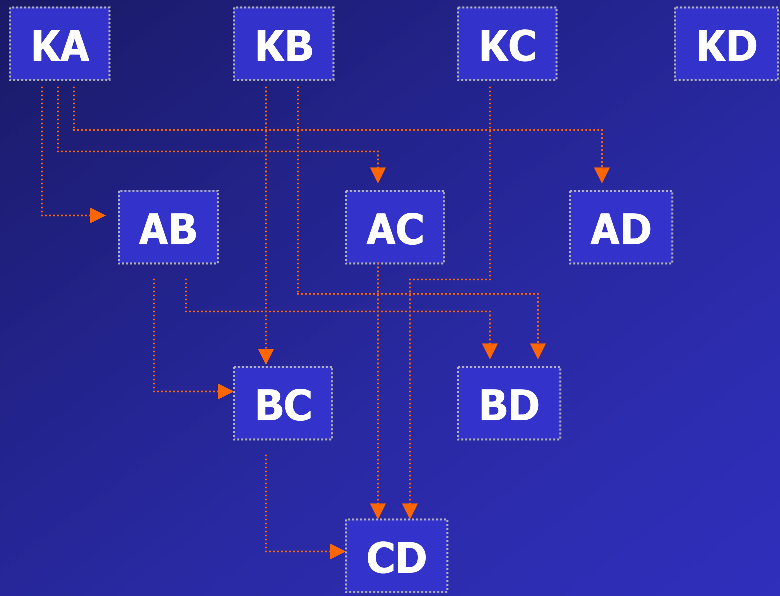
A	B
a_3	b_2
a_7	b_9
a_9	b_3
…	…
a_7	b_4

How: Embed a watermark *in all expected partitions* (closure).

mark interference

K	A	B	C	D
…	…	…	…	…

Question: How to deal with multiple marks encoding interference ?



How: Maintain a mark interference dependency graph so as to *propagate awareness of changes*.

attack: “bucket counting”

A	B
a_2	b_3
a_1	b_7
…	
a_i	$b_{f(i)}$
…	
a_2	b_8
a_1	b_7

Question: How to survive a correlation attack aimed at detecting statistical bias in case of non-primary key first argument ?

problem: values in **A** (non-key) can repeat

$\rightarrow$ Mallory can “count buckets” for $[a_i,b_{f(i)}]$ pairs and identify “hot spots”

solution: increase size of possible assigned target values for **B** when encoding.

$f(i) = msb(H(a_i,k),log_2(n_A))$

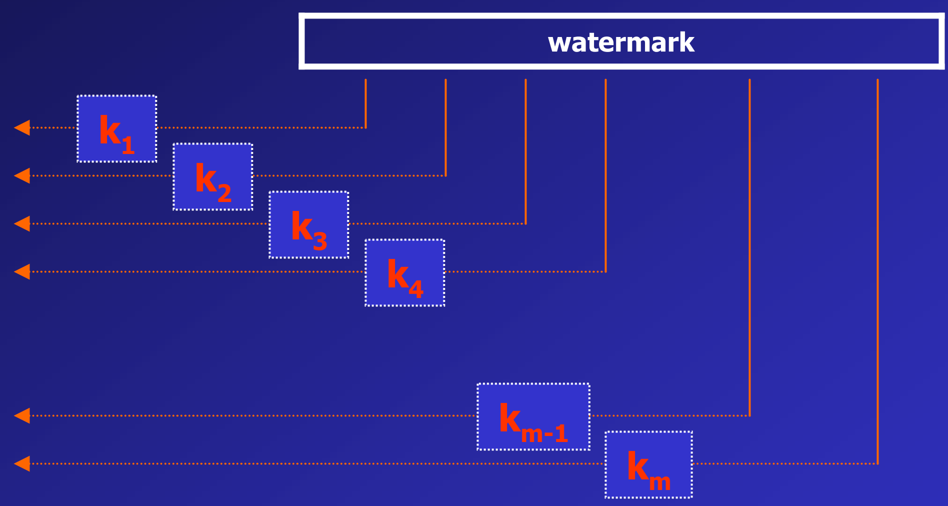
$f(i) \in \{msb(H(a_i,x),log_2(n_A)) | x \in \{k_1,k_2,...\}\}$

How: Larger target value sets for fit tuples.

attack: “bucket counting” revisited

Question: How to survive a correlation attack aimed at detecting statistical bias in case of non-key first argument ?

A	B
a_2	b_3
a_1	b_7
…	
a_i	$b_{f(i)}$
…	
a_2	b_8
a_1	b_7



global key $\rightarrow$ $\mathbf{k=(k_1,k_2,\ldots,k_m)}$

low-impact encoding

How: Spread correlation among multiple layers of weak(er) “low-impact” encodings.

attack: extreme partitioning

Question: How to survive extreme, single-attribute partitioning ?

A
a_3
a_7
a_9
a_7

question: what is of
val

one answer: frequency domain of value occurrences

K	A	B
0	a_3	b_2
1	a_7	b_9
2	a_9	b_3
...		...
n	a_7	b_4

B
b_2
b_9
b_3
...
b_4

solution: alter this frequency domain value occurrence histogram to encode an additional watermark.

How: Frequency domain embedding.

attack: value re-mapping

Question: How to deal with bijective value re-mappings ?

$A \in \{a_1,\ldots,a_{n_A}\}$

K	A
0	a_3
1	a_7
2	a_9
…	
n	a_7

$\text{map}(a_i)=a'_i$

K	A'
0	a'_3
1	a'_7
2	a'_9
…	
n	a'_7

$A' \in \{a'_1,\ldots,a'_{n_A}\}$

$$\text{freq}(a_i) \cong \text{freq}(a'_i) \rightarrow \text{detect map}^{-1}(a_i)$$

How: Discover inverse mapping by using frequency histograms.

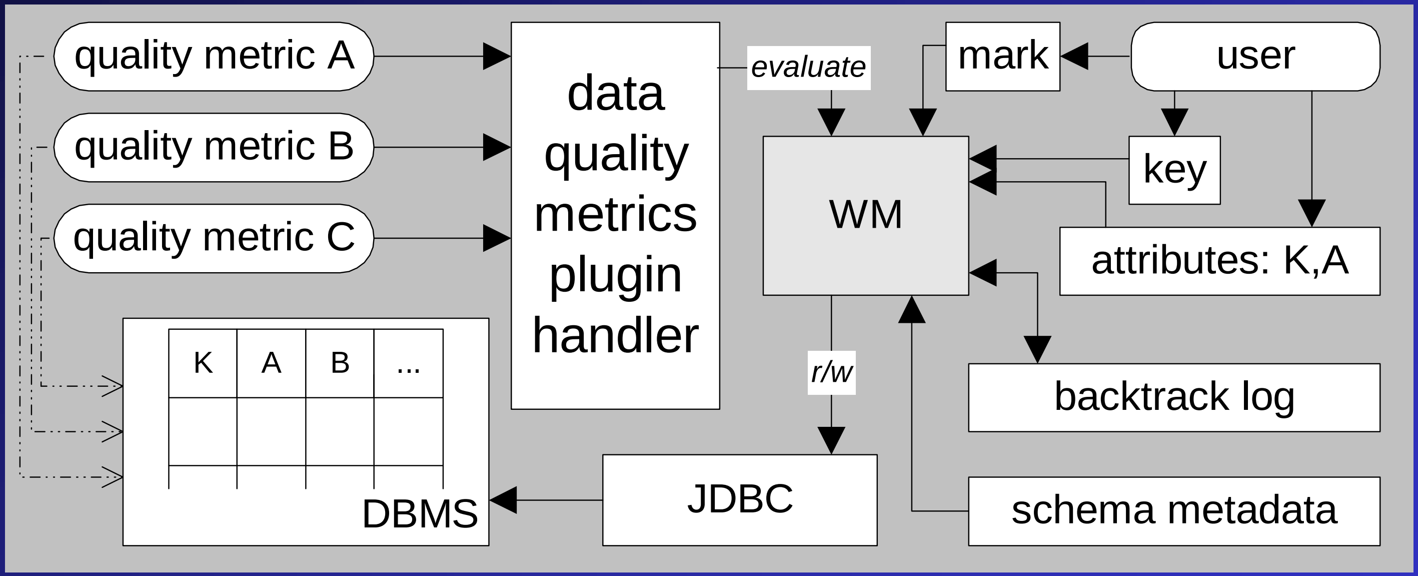
attack: informed inverts

Question: How to survive informed mark removal attacks ?

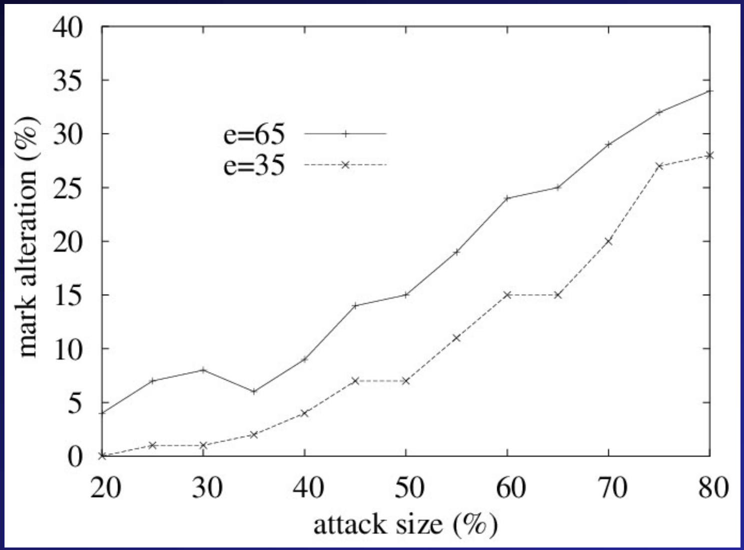
solution: if Mallory inverts embedding **2** it effectively enforces **1** (*collision set* bias).

How: Multiple self-reinforcing layers.

architecture

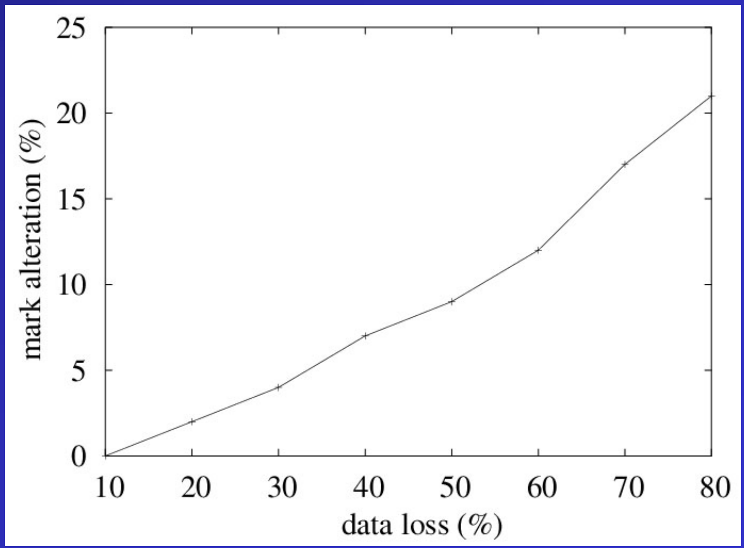


selected results



watermark recovery in the presence of random alterations (error corrected)

watermark recovery in the presence of data loss (error corrected)



resilience analysis

We ask: what is the probability of altering *at least* ***r*** *bits* in the detected mark *by an attack of size* ***a*** ?

$$P(r,a) = P'(r, \frac{a}{e}) = \sum_{i=r}^{a/e} \binom{a/e}{i} \times p^i \times (1-p)^{(a/e)-i}$$

This looks like binomial sampling with $P(X_i=1)=p$ and $P(X_i=0)=1-p$. Thus $P(\Sigma X_i>r)$ (at least r bits) can be re-cast as $P(f(\Sigma X_i)>f(r))$.

$$f(\sum X_i) = \frac{\sum X_i - mean}{stdev} = \frac{\sum X_i - \frac{a}{e} \times p}{\sqrt{\frac{a}{e} \times p \times (1-p)}}$$

But $f(\Sigma X_i)$ behaves aprox. normal $N(0,1)$ (central limit) and we know $f(r)$. Now we can estimate $P(f(\Sigma X_i)>f(r))$ by table lookup:

$P(15,1200)=31.6\%$

categorical data: analysis

In other words if Mallory alters *20%* of the data (*1200*) with a *70%* success rate for each bit flip, the probability that he succeeds in destroying *15* mark data bits (before error correction) is *31%.*

If we consider error correction tolerating $t_{ecc}=5\%$ errors, in reality Mallory alters only a fraction of the $r=15$ bits in the corrected mark:

$$\left(\frac{r \times e}{N} - t_{ecc}\right) \times \frac{|wm|}{|wm\_data|} = 1\%$$

That is just *one bit*. In other words, if Mallory wants to alter a single error corrected bit with a *31%* likelihood, he needs to modify at least *20%* of the data ! But in doing so he is likely to *destroy much the value* of the data !

categorical data: analysis

We ask: how many watermarking alterations are *required* to guarantee a given *upper bound* on *one bit alteration attack* vulnerability ?

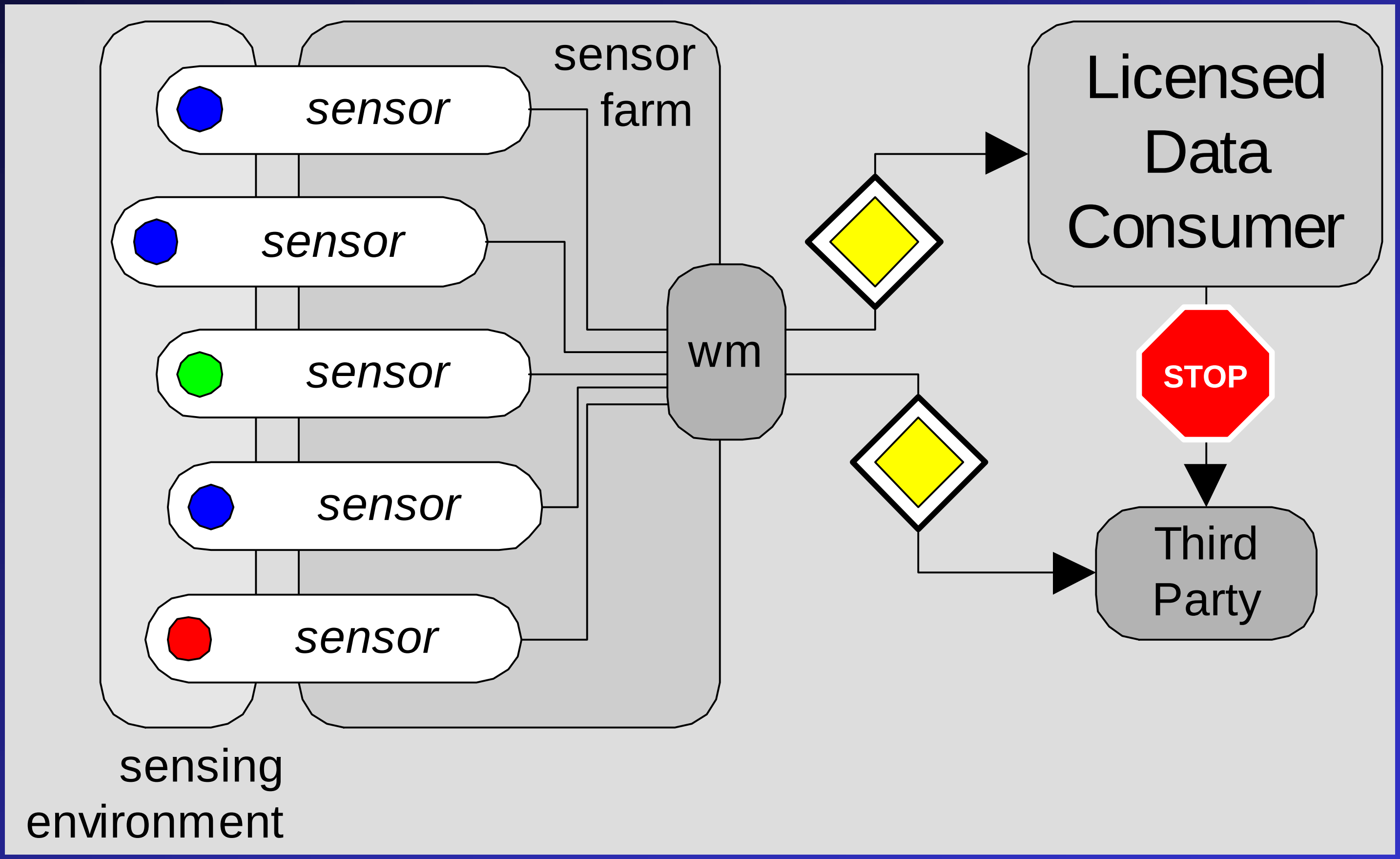
Assume: Mallory cannot afford to modify more than *20%* of the data and we desire an attack vulnerability **<** *10%*.

It can be shown that we have to alter *only 2.1%* of the tuples to guarantee an upper bound of *10%*.

talk pointer

- introduction
- existing research: media
- beyond media
- numeric relational data
- categorical data
- → **sensor streams**
- limits of watermarking
- the future

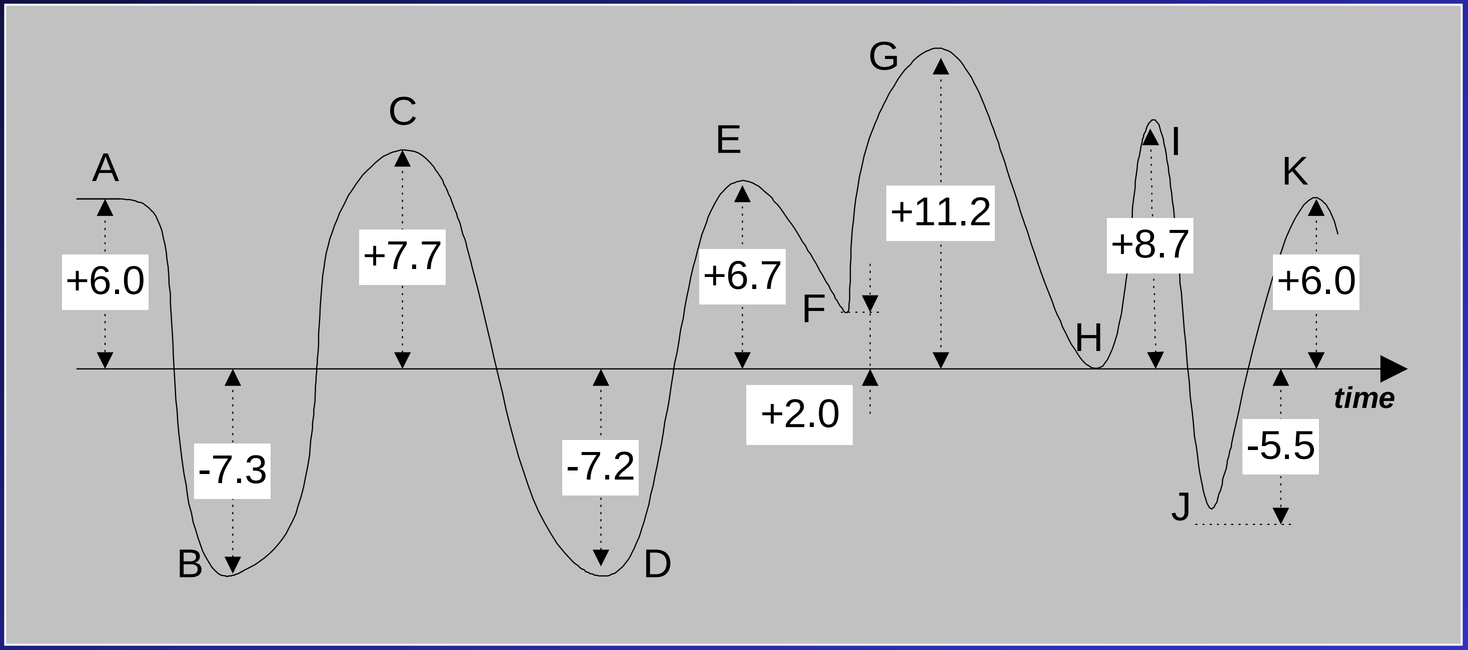
sensor streams: scenario



sensor streams: challenges

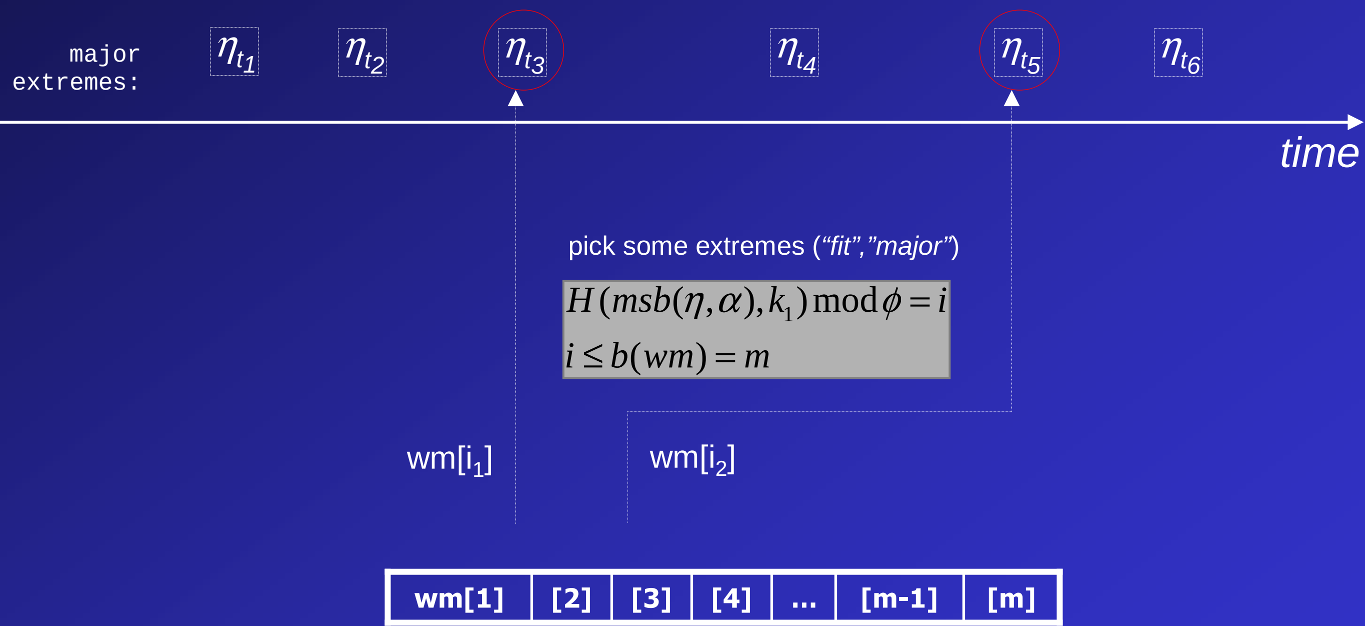
- Transforms
  - → Segmentation (no timestamps !)
  - → Summarization
  - → Sampling
- Attacks
  - → Random Alterations
  - → Linear Changes
- Streaming model
  - → Fast encoding (almost real time)
  - → Single-pass (no second look at data)
  - → Memory bounds

sample temperature stream

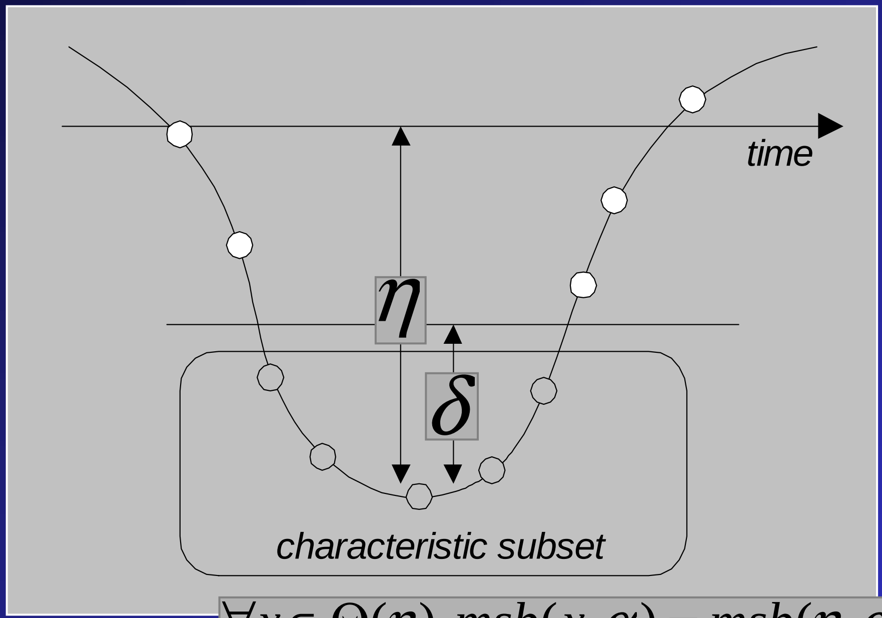


trivial embedding (no timeline)

idea: *use "resilient" stream "features" to encode watermark bits (e.g., extremes)*



trivial characteristic subset encoding



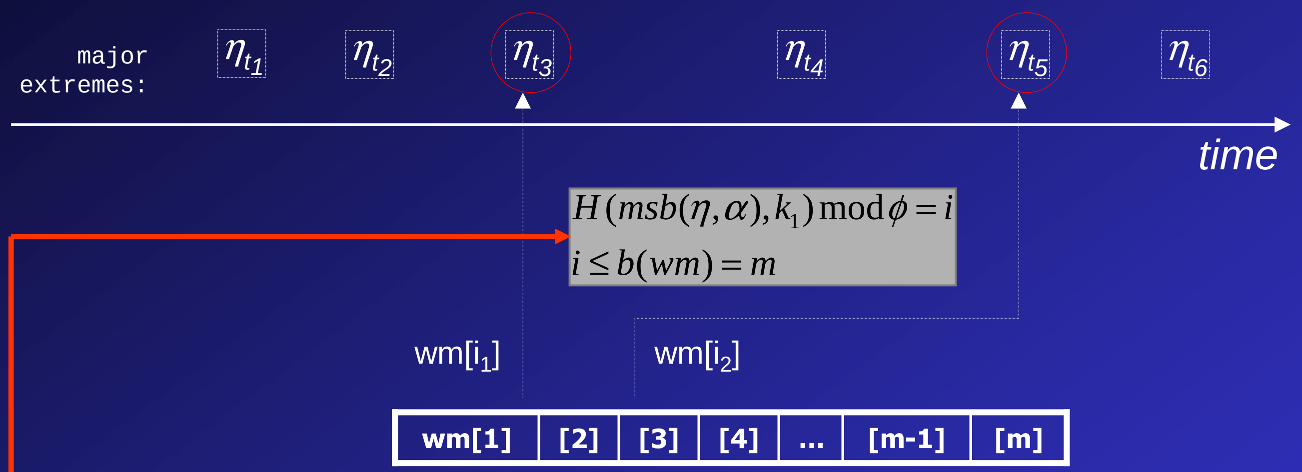
$$\forall x \in \Theta(\eta), msb(x,\alpha) = msb(\eta,\alpha)$$

Encoding:

$$\forall x \in \Theta(\eta), x[bit] = wm[i],$$
$$bit = H(msb(\eta,\alpha), k_1) \bmod \beta$$

→ **deals with:** sampling, summarization

vulnerability: correlation detection

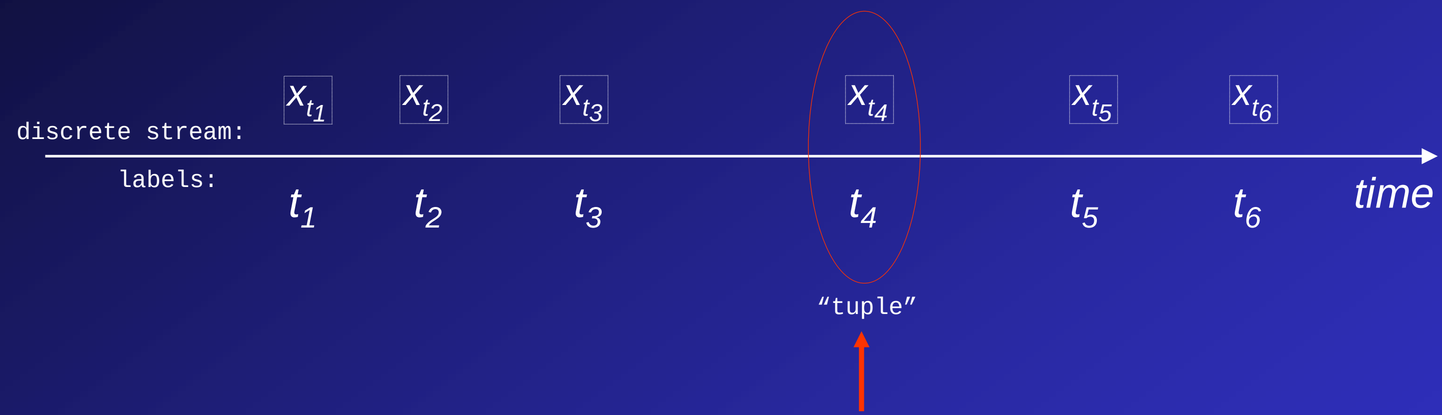


problem: bit-location-MSB correlation

values in MSB can repeat → Mallory can “count buckets” per individual unique MSB values and identify “hot spots”

→ **need:** alternate source of (pseudo-) randomness

if we would have a reference timeline …



*… then we could maybe apply existing relational data methods (e.g. numeric watermarking by Kiernan and Agrawal, VLDB 2002) – would survive random alterations (but **not** summarization and other transforms).*

“timeline” requirements

• survives:

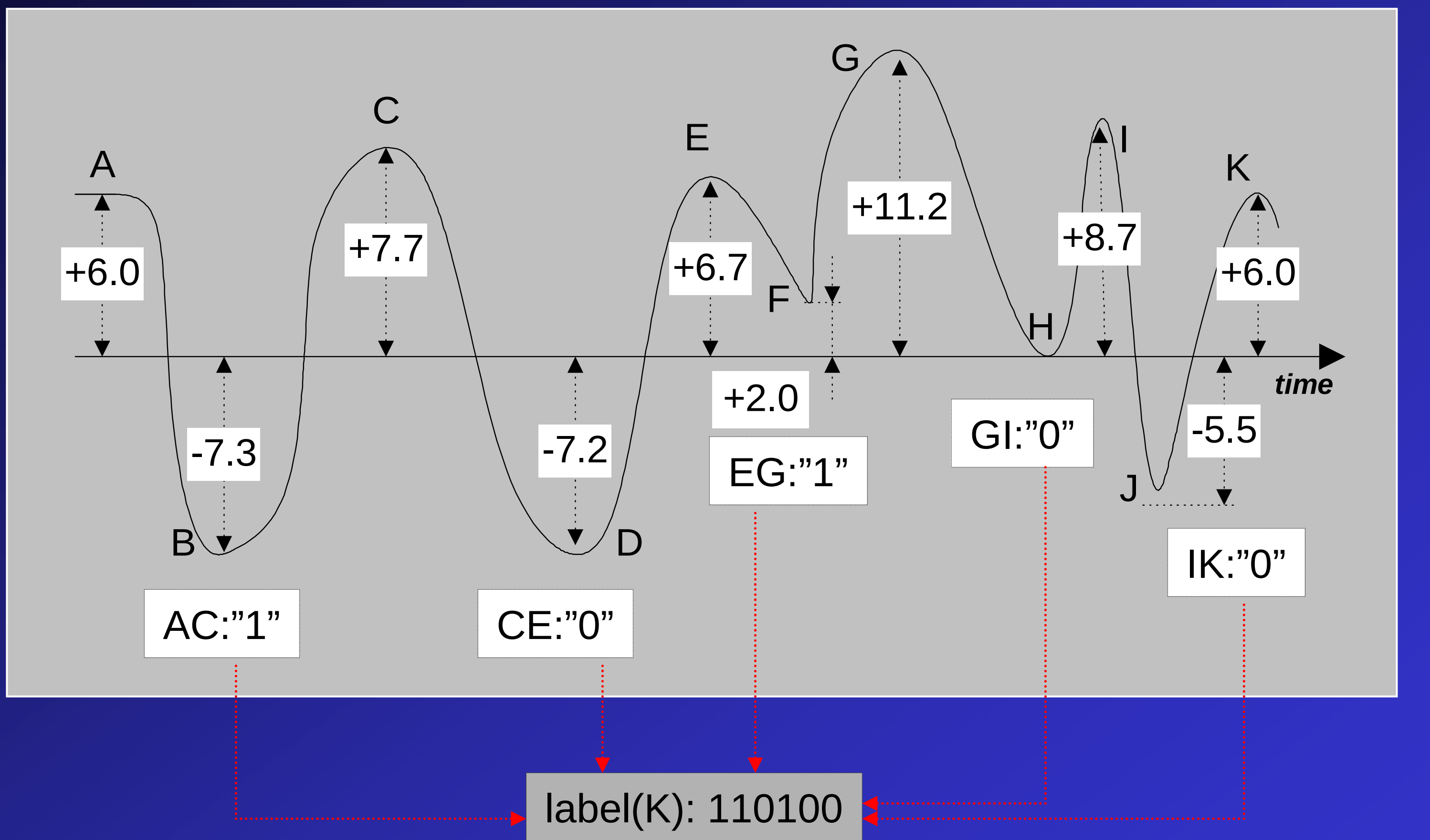
→ transforms (summarization, sampling)
→ attacks (segmentation, random alterations)

• can be constructed:

→ fast
→ from little more than a window of data

constructing “timeline”: stream behaviour

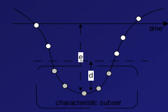
idea: *timeline = “differential interpretation” of stream behavior*



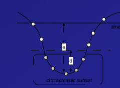
→ **deals with:** segmentation, sampling, summarization

bit embedding (with timeline)

$bit = H(t_3, k_1) \bmod \beta$



$bit = H(t_5, k_1) \bmod \beta$



major extremes: η_{t_1} η_{t_2} η_{t_3} η_{t_4} η_{t_5} η_{t_6}

labels: t_1 t_2 t_3 t_4 t_5 t_6 *time*

$$H(msb(\eta, \alpha), k_1) \bmod \phi = i$$
$$i \leq b(wm) = m$$

wm[i_1] wm[i_2]

wm[1]	**[2]**	**[3]**	**[4]**	**...**	**[m-1]**	**[m]**

partial problem remains:

→ Mallory can "count buckets" per individual unique MSB values (+previous labels!) and *sometimes* identify "hot spots"

→ **smarter:** labeling, encoding convention

smarter characteristic subset encoding

"partial hash sums"

$$\Theta(\eta,\delta) = \{x_1,\ldots,x_a\}$$

$$\forall i \le j \in [1,a], m_{ij} = \frac{\sum_{u\in[i,j]} x_u}{j-i+1}$$

encoding convention:

"true"

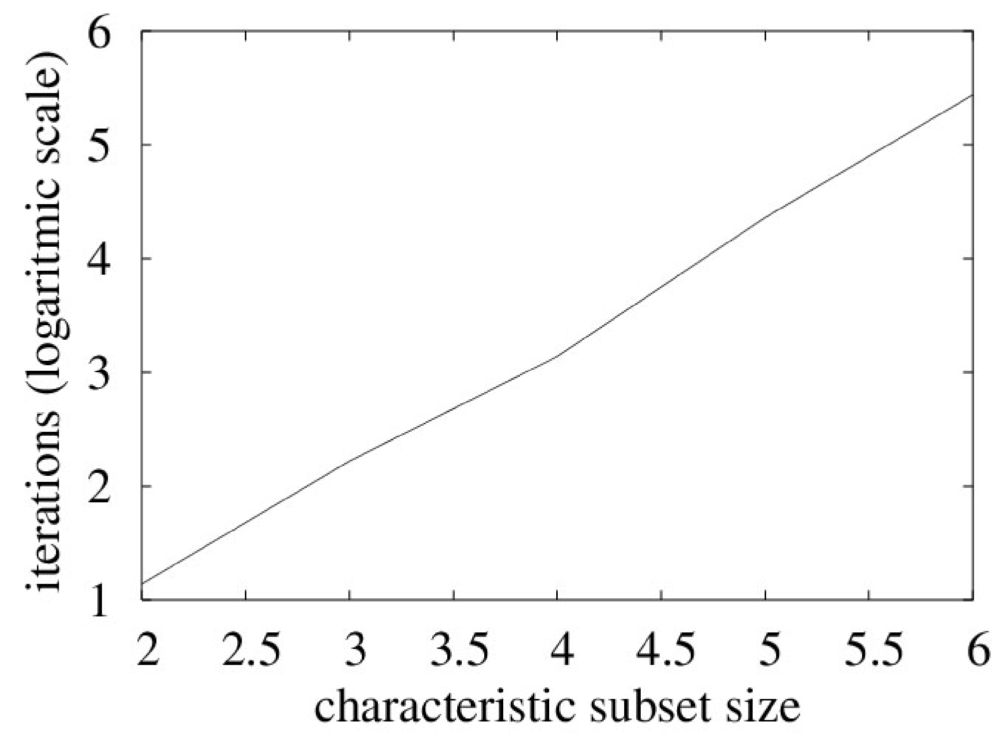
$$lsb(H(lsb(m_{ij},\beta), label(\eta)), \zeta) = 2^{\zeta} - 1$$

"false"

$$lsb(H(lsb(m_{ij},\beta), label(\eta)), \zeta) = 0$$

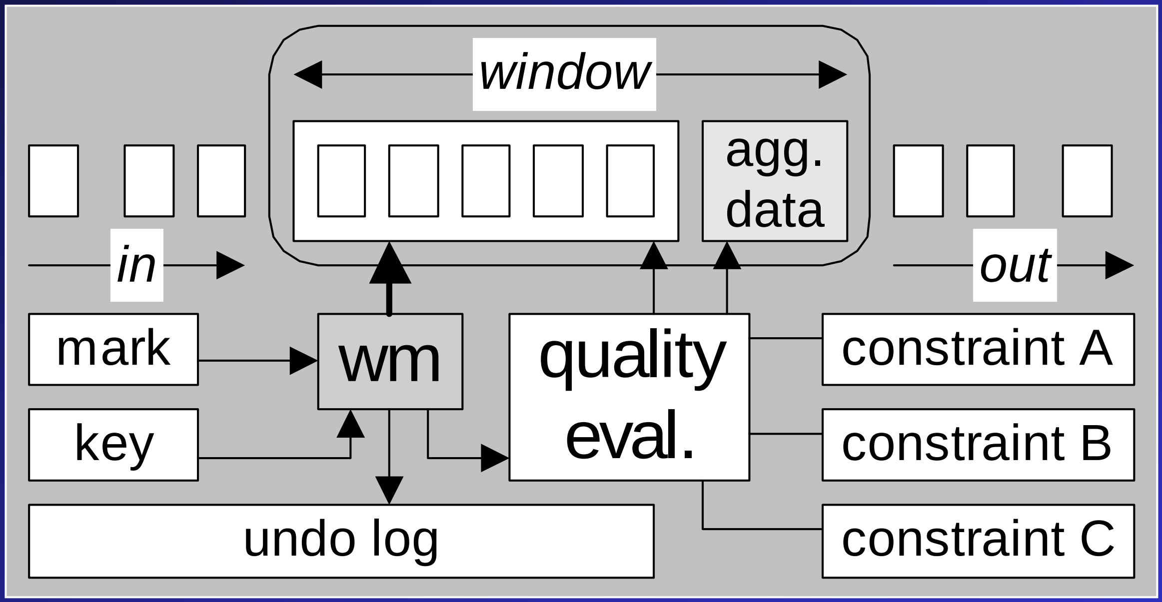
- pros:
  - → survives summarization
  - → provides randomness
  - → de-correlates encoding location
- cons:
  - → finding conforming data point is computationally very expensive

generate data with conforming partial sums

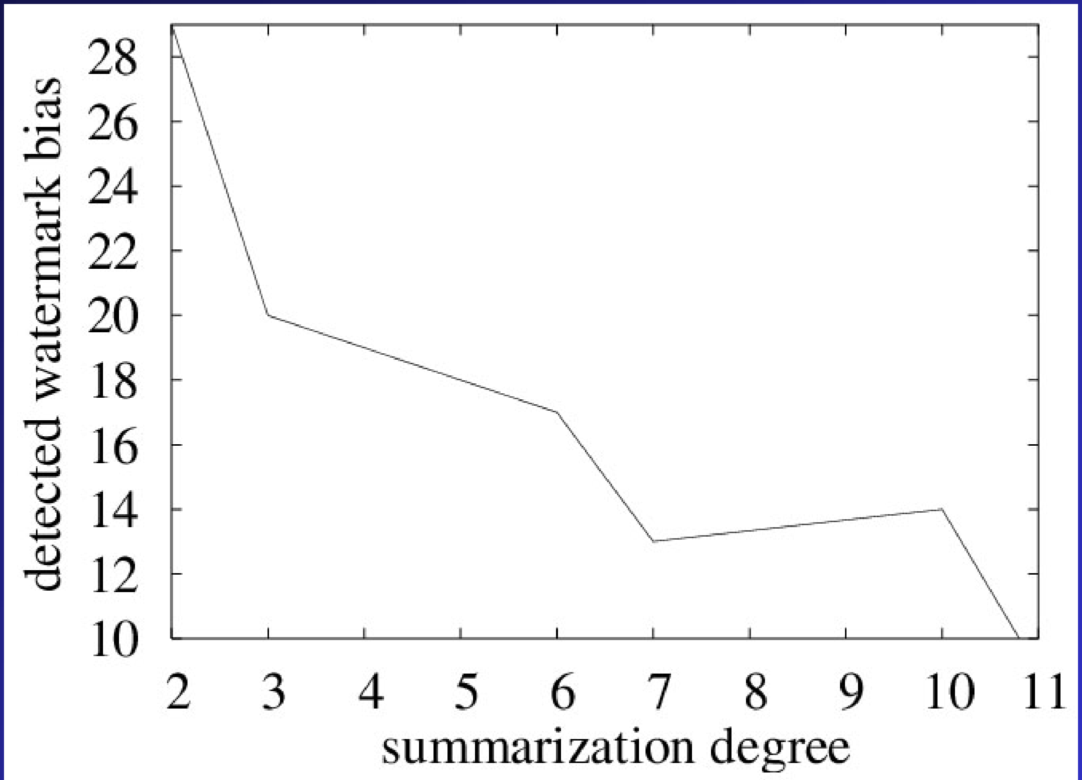


computation required is exponential, but we are ok because we can likely get away with computing just a few of them

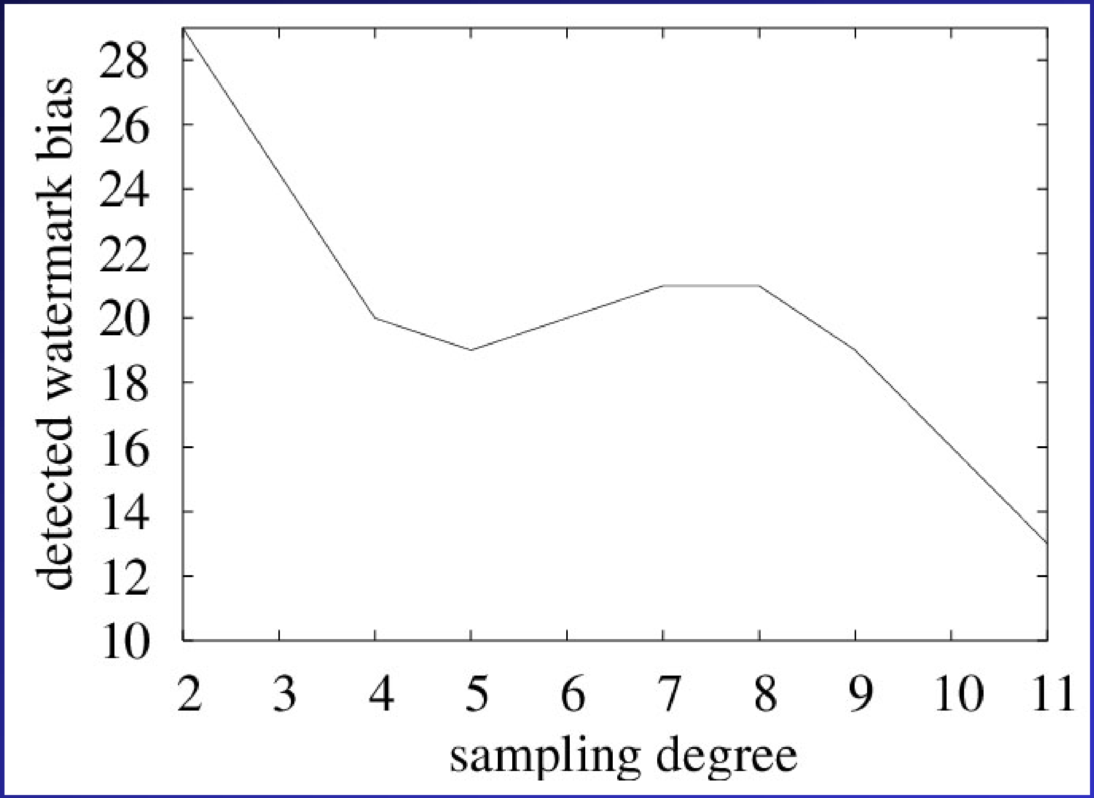
sensor streams: implementation



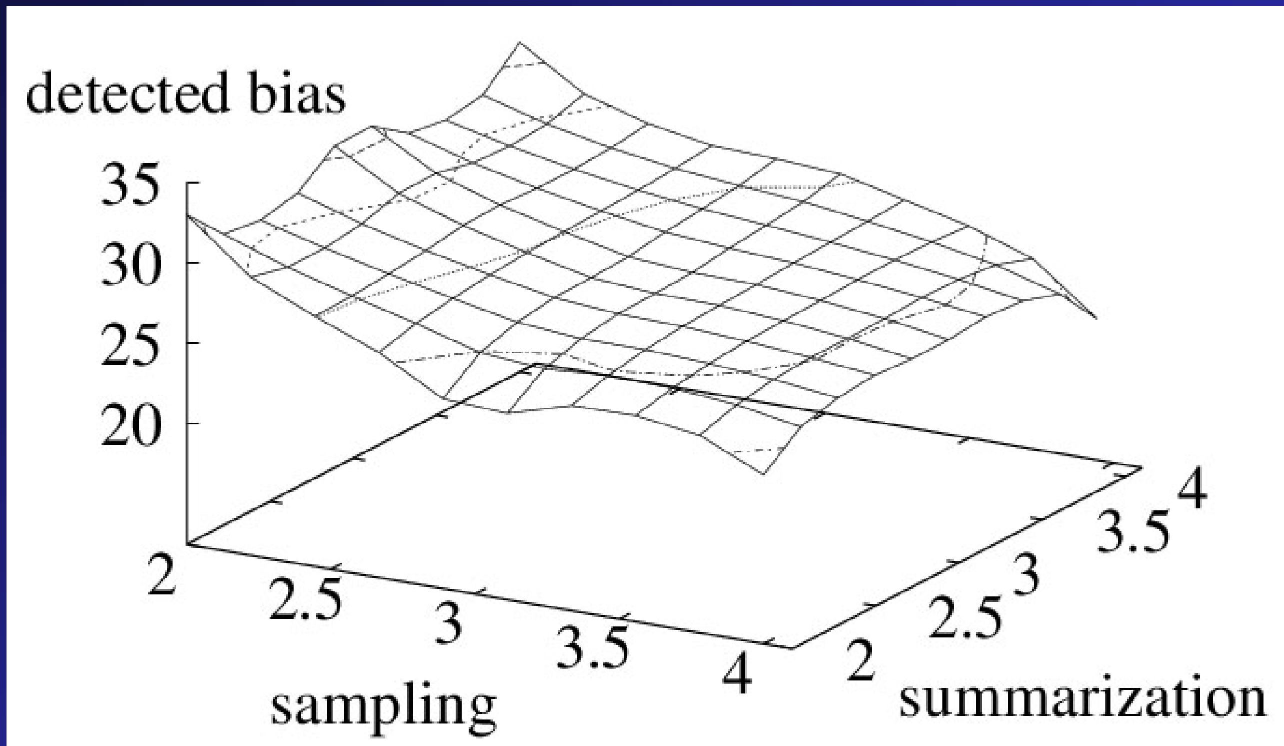
sensor streams: summarization



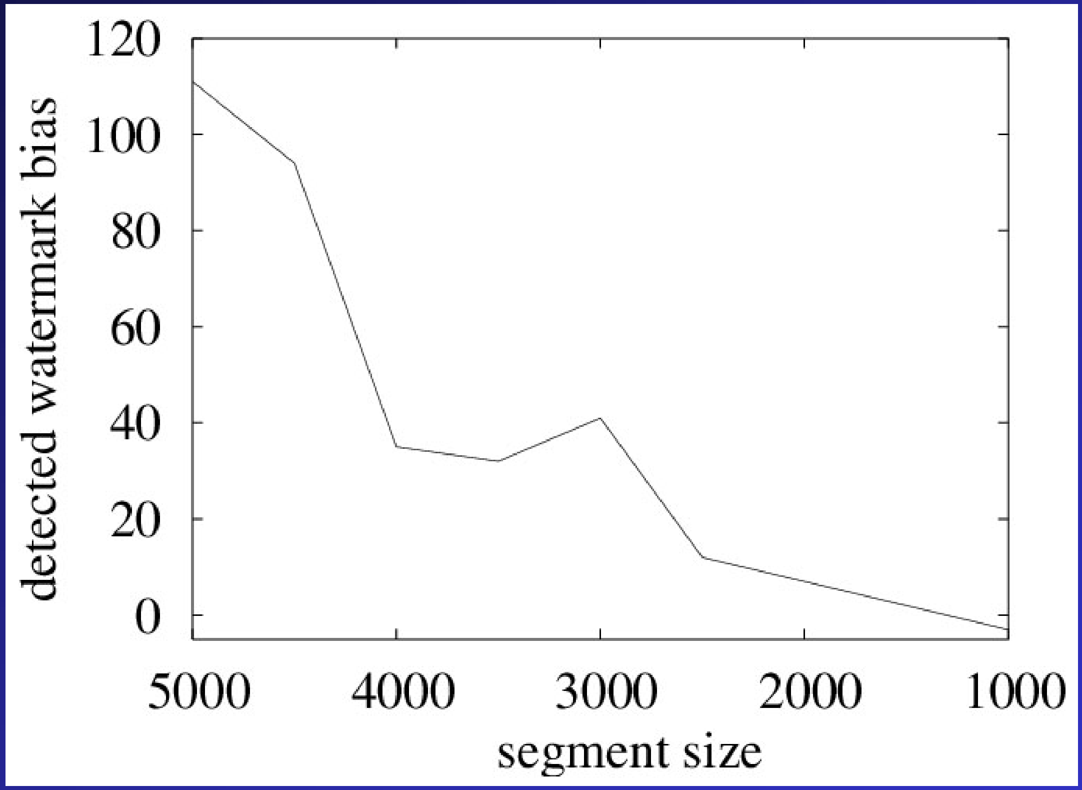
sensor streams: sampling



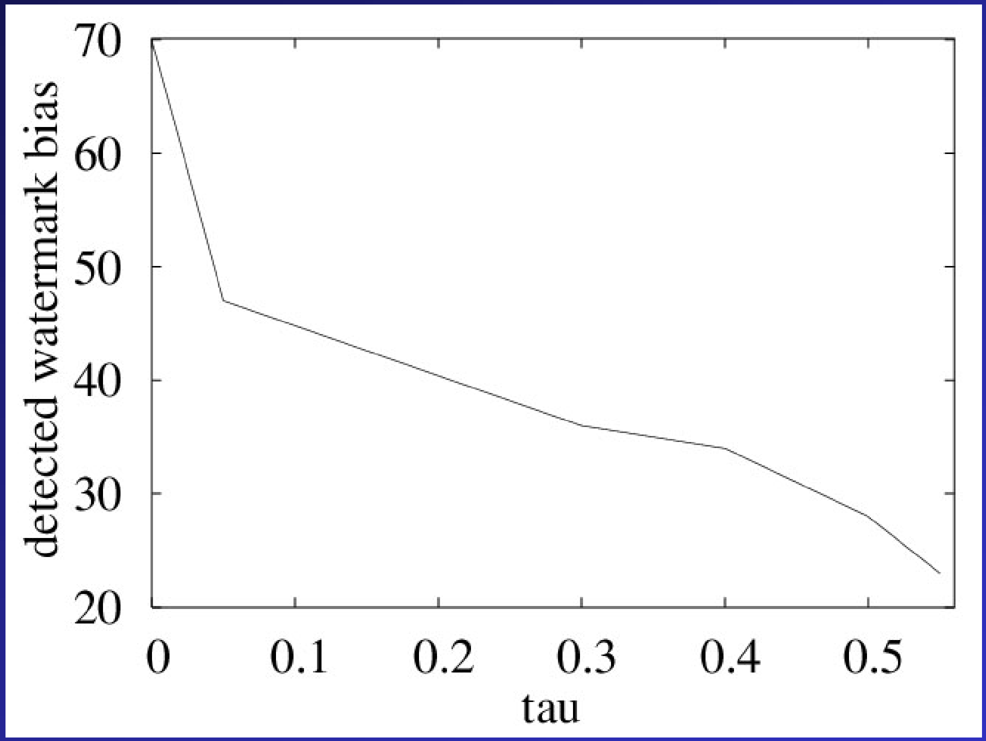
sensor streams: sampling+summarization



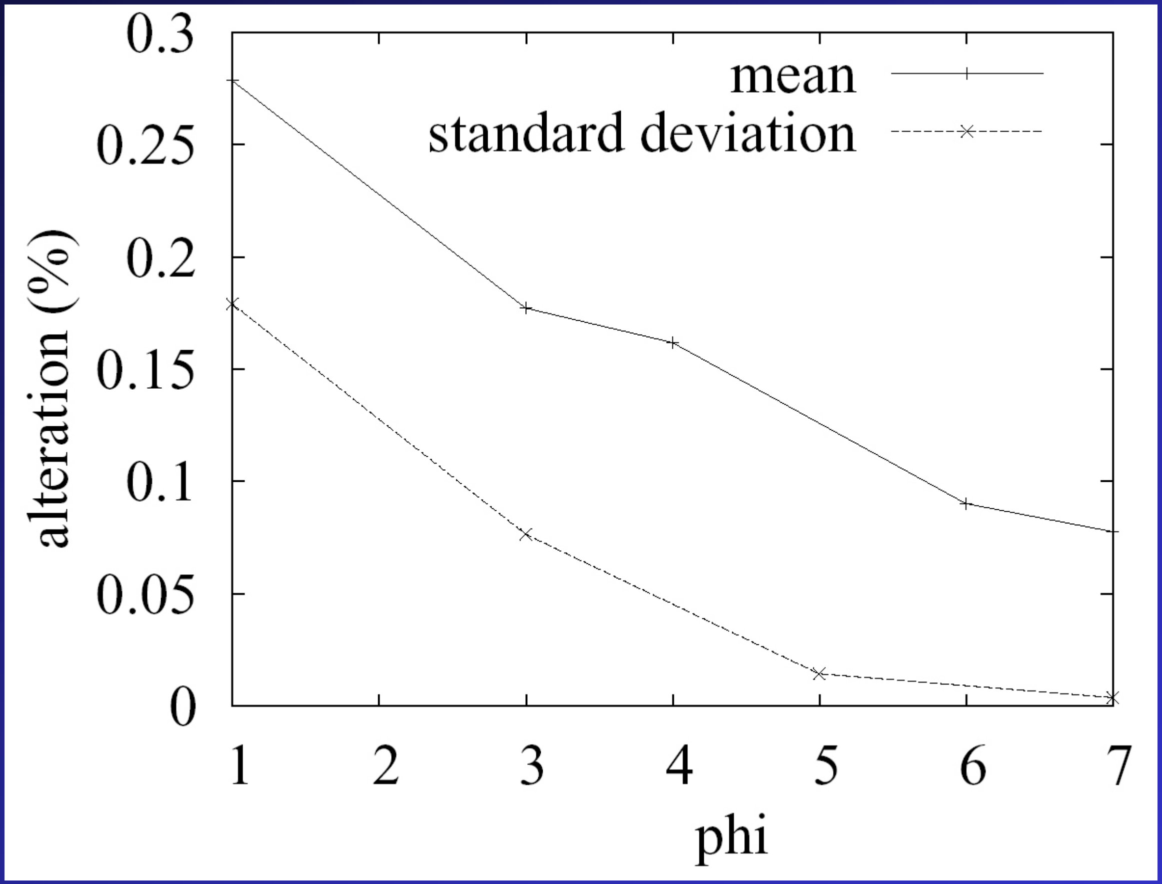
sensor streams: segmentation



sensor streams: random alteration



sensor streams: impact on quality



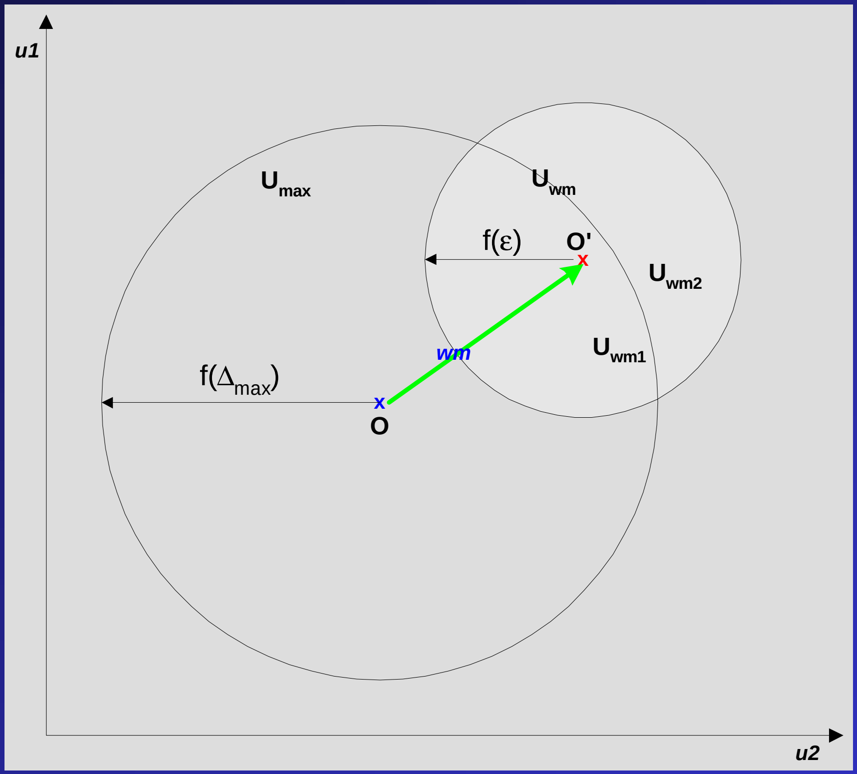
talk pointer

introduction
existing research: media
beyond media
numeric relational data
categorical data
sensor streams
→ limits of watermarking
the future

limits

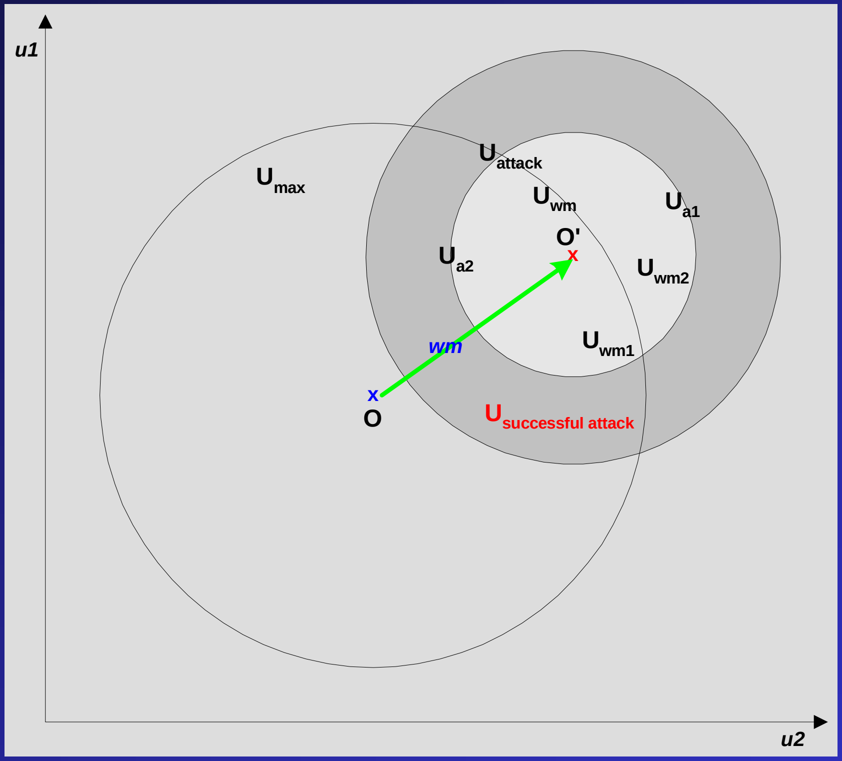
Are there any bounds one can assess for watermarking as a tool for rights protection ?

limits: usability spaces



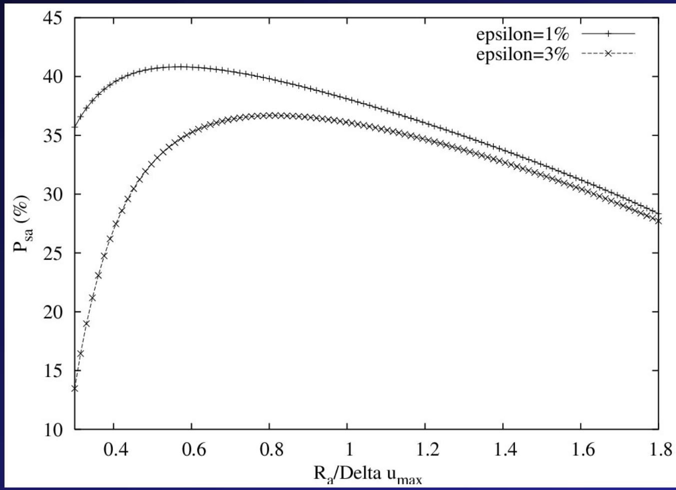
A point uniquely identifies a Work in **D** in this 2 dimensional view of an usability space. Watermarking is a translation that results in O', a watermarked version of O.

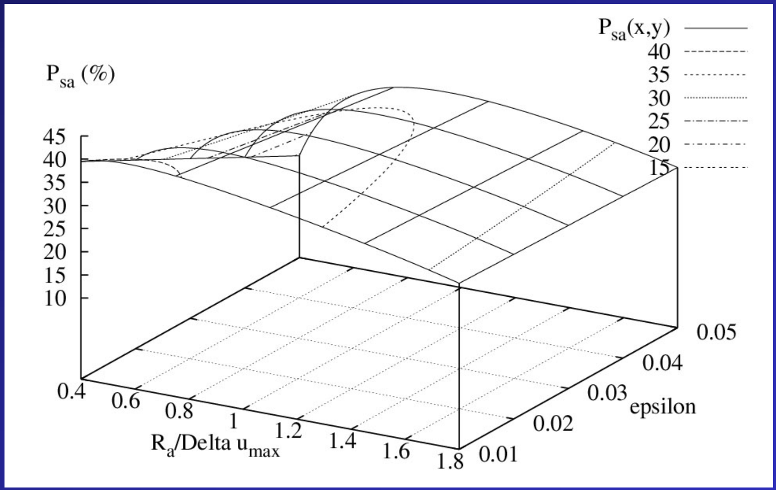
limits: Mallory attacks



A successful attack is one that yields results in the area of intersection between U_{max} and $(U_a - U_{wm})$.

limits: inherent vulnerability





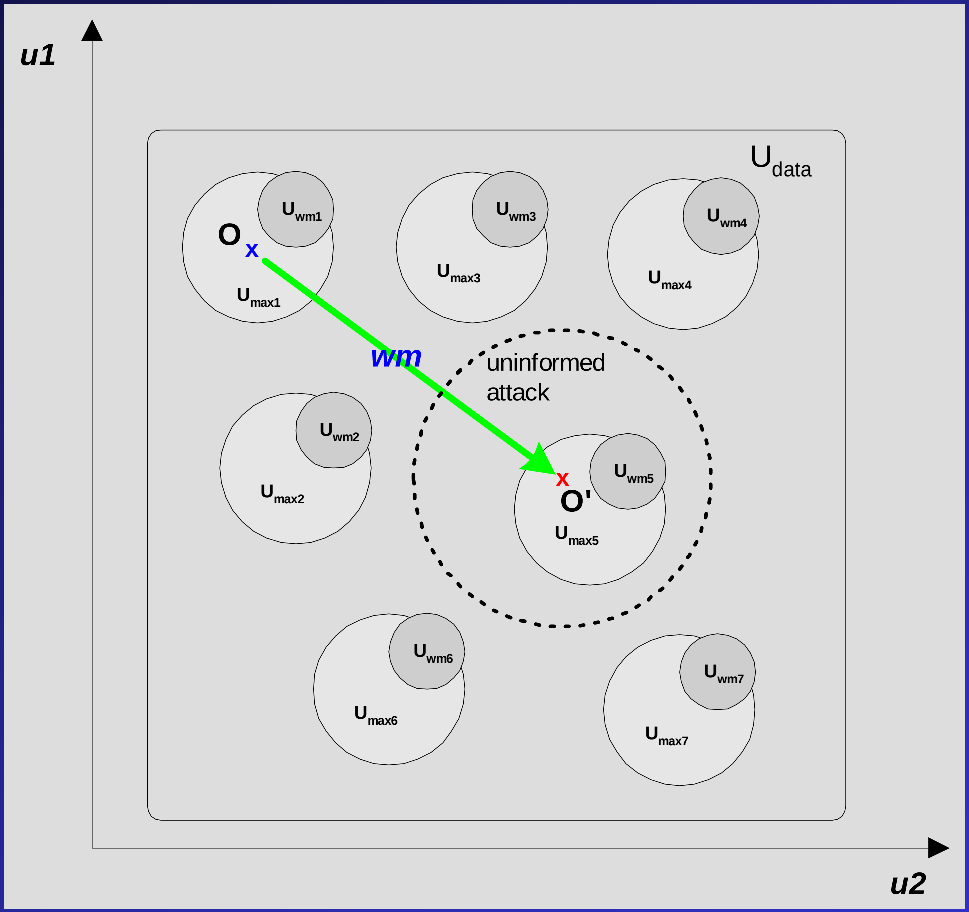
No matter what the watermarking method, there exists a random attack with a significant non-zero success probability. A more convincing mark yields an even more vulnerable bound on attack success probability.

$$P_{sa} = \frac{\| U_{a2} \| - \varepsilon_w \pi \Delta u_{\max}^2}{\pi R_a^2} = \frac{d^2 \cos^{-1}(1 - \frac{R_a^2}{2d^2}) + R_a^2 \cos^{-1}(\frac{R_a}{2d}) - \frac{1}{2} R_a \sqrt{4d^2 - R_a^2} - \varepsilon_w \pi d^2}{\pi R_a^2}$$

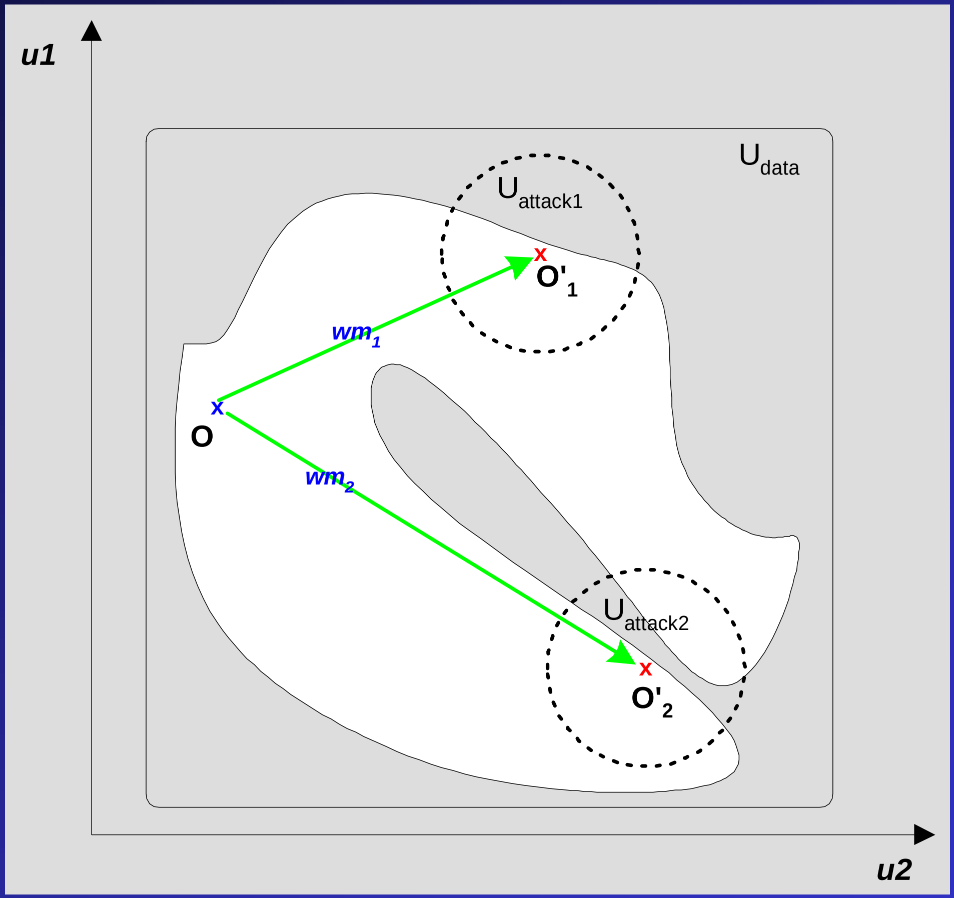
limits: what if scenarios

- high dimensionality
- sparse vicinities
- concavity
- smart Mallory

limits: sparse vicinities



limits: concave shapes



watermarking principles

Convince-ability Trade-off.

There exists a direct relationship between the probability of success of a random attack and the ability to convince in court. The more convincing in court, the higher the probability of success of a random attack.

→ *are there classes for which this is not true ?*

“Optimality” Principle.

The vulnerability of a watermarking scheme is minimized when it yields watermarked result Works on the boundary of the maximum allowable distortion vicinity of the originals.

→ *recommendation for algorithm design*

talk pointer

introduction
existing research: media
beyond media
numeric relational data
categorical data
sensor streams
limits of watermarking
→ the future

future

→ knowledge centric model of security attacks

→ integrate constraint handling in encoding

→ multi-source data integration

→ extend limit proofs, understand broader class

→ optimizer: find sweet spot in encoding space

→ wmdb.*: backtrack pruning speed up

→ protect categorical data streams

→ intersection: categorical - numerical data

→ alteration distance (categorical data)

selected references

[1] Rakesh Agrawal, Peter J. Haas, and Jerry Kiernan. Watermarking relational data: framework, algorithms and analysis. The VLDB Journal, 12(2):157-169, 2003.

[2] David Gross-Amblard. Query-preserving watermarking of relational databases and xml documents. In Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pages 191-201, New York, NY, USA, 2003. ACM Press.

[3] J. Kiernan and R. Agrawal. Watermarking relational databases. In Proceedings of the 28th International Conference on Very Large Databases VLDB, 2002.

[4] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. Constructing a virtual primary key for fingerprinting relational data. In DRM '03: Proceedings of the 2003 ACM workshop on Digital rights manage ment, pages 133-141, New York, NY, USA, 2003. ACM Press.

[5] Radu Sion. Proving ownership over categorical data. In Proceedings of the IEEE International Conference on Data Engineering ICDE, 2004.

[6] Radu Sion. wmdb.*: A suite for database watermarking (demo). In Proceedings of the IEEE International Conference on Data Engineering ICDE, 2004.

[7] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Rights protection for relational data. In Proceedings of the ACM Special Interest Group on Management of Data Conference SIGMOD, 2003.

[8] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Relational data rights protection through watermarking. IEEE Transactions on Knowledge and Data Engineering TKDE, 16(6), June 2004.

[9] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Resilient rights protection for sensor streams. In Proceedings of the Very Large Databases Conference VLDB, 2004.

[10] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Ownership proofs for categorical data. IEEE Transactions on Knowledge and Data Engineering TKDE, 2005.

eof

Thank You !

more references

[1] Rakesh Agrawal, Peter J. Haas, and Jerry Kiernan. Watermarking relational data: framework, algorithms and analysis. The VLDB Journal, 12(2):157-169, 2003.

[2] David Gross-Amblard. Query-preserving watermarking of relational databases and xml documents. In Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pages 191-201, New York, NY, USA, 2003. ACM Press.

[3] J. Kiernan and R. Agrawal. Watermarking relational databases. In Proceedings of the 28th International Conference on Very Large Databases VLDB, 2002.

[4] Yingjiu Li, Huiping Guo, and Sushil Jajodia. Tamper detection and localization for categorical data using fragile watermarks. In DRM '04: Proceedings of the 4th ACM workshop on Digital rights management, pages 73-82, New York, NY, USA, 2004. ACM Press.

[5] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. Constructing a virtual primary key for fingerprinting relational data. In DRM '03: Proceedings of the 2003 ACM workshop on Digital rights manage ment, pages 133-141, New York, NY, USA, 2003. ACM Press.

[6] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. A robust watermarking scheme for relational data. In Proceedings of the Workshop on Information Technology and Systems (WITS), pages 195-200, 2003.

[7] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. Defending against additive attacks with maximal errors in watermarking relational databases. In Proceedings of the IFIP WG 11.3 Working Conference on Data and Application Security, pages 81-94, 2004.

[8] Yingjiu Li, Vipin Swarup, and Sushil Jajodia. Fingerprinting relational databases: Schemes and specialties. IEEE Transactions on Dependable and Secure Computing, 2(1):34-45, 2005.

[9] Radu Sion. Proving ownership over categorical data. In Proceedings of the IEEE International Conference on Data Engineering ICDE, 2004.

[10] Radu Sion. Rights Assessment for Discrete Digital Data, Ph.D. dissertation. Computer Sciences, Purdue University, 2004.

[11] Radu Sion. wmdb.*: A suite for database watermarking (demo). In Proceedings of the IEEE International Conference on Data Engineering ICDE, 2004.

[12] Radu Sion and Mikhail Atallah. Attacking digital watermarks. In Proceedings of the Symposium on Electronic Imaging SPIE, 2004.

[13] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. On watermarking numeric sets. Online at https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/, 2001.

[14] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. On watermarking numeric sets. In Proceedings of IWDW 2002, Lecture Notes in Computer Science. Springer-Verlag, 2002.

[15] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Power: Metrics for evaluating watermarking algorithms. In Proceedings of IEEE ITCC 2002. IEEE Computer Society Press, 2002.

[16] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Watermarking databases. Online at https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/, 2002.

[17] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Rights protection for relational data. In Proceedings of the ACM Special Interest Group on Management of Data Conference SIGMOD, 2003.

[18] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Relational data rights protection through watermarking. IEEE Transactions on Knowledge and Data Engineering TKDE, 16(6), June 2004.

[19] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Resilient rights protection for sensor streams. In Proceedings of the Very Large Databases Conference VLDB, 2004.

[20] Radu Sion, Mikhail Atallah, and Sunil Prabhakar. Ownership proofs for categorical data. IEEE Transactions on Knowledge and Data Engineering TKDE, 2005.

David-Gross Amblard, PODS 2003

The main difficulty preserving query results “is linked to the informational complexity of sets defined by queries, rather than their computational complexity". Roughly, if the family of sets defined by the queries is not learnable [36], no query-preserving data alteration scheme can be designed.

Under certain assumptions (i.e., query sets defined by first-order logic and monadic second order logic on restricted classes of structures – with a bounded degree for the Gaifman graph or the tree-width of the structure) a query-preserving data alteration scheme indeed exists.

relational data: scenario

Given a numeric relational database **B(a_1 ,…, a_n)**, a set of local and global semantic constraints **C**, and a set of secrets **K**, determine a watermarked version **B'(a'$_1$, …, a'$_n$)** of **B**, such that all elements in **C** are satisfied (over **B'**) and **B'** features enough watermark *resilience*.

example challenges: what is "*resilience*" *?* recover mark with minimal context (i.e. no original data available) ?