Watermarking Relational Databases

Acknowledgement: Mohamed Shehab from Purdue Univ.

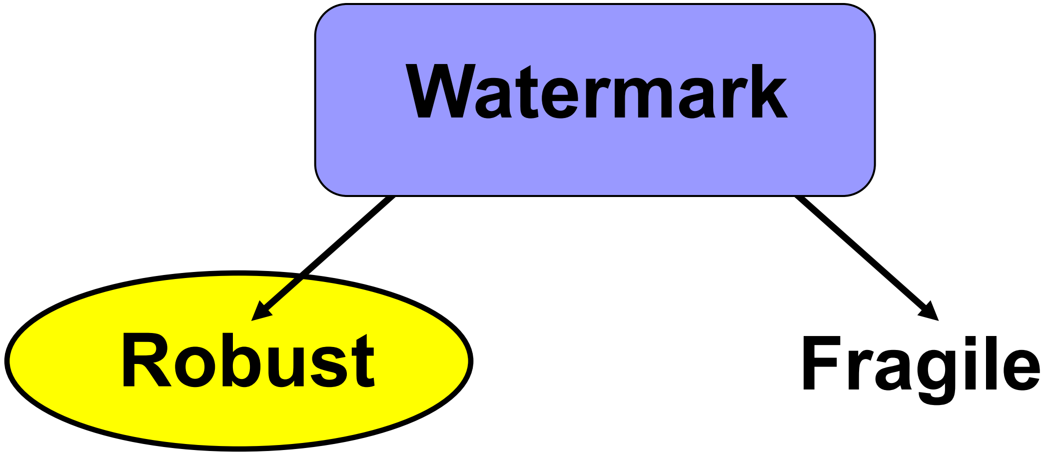
Outline

- Introductory Material
- General Watermarking Model & Attacks
- WM Technique 1 (Agrawal et al.)
- WM Technique 2 (Sion et al.)
- Future Challenges and References

What is Watermarking ?

- A *"watermark"* is a signal that is securely, imperceptibly, and *"robustly"* embedded into original content such as an image, video, or audio signal, producing a watermarked signal.
- The watermark describes information that can be used for proof of ownership or tamper proofing.

What is Watermarking ? (Cont.)



- Robust Watermark: for proof of ownership, copyrights protection.
- Fragile Watermark: for tamper proofing, data integrity.

Why Watermarking ?

- Digital Media (Video, Audio, Images, Text) are easily copied and easily distributed via the web.
- Database outsourcing is a common practice:
  - Stock market data
  - Consumer Behavior data (Walmart)
  - Power Consumption data
  - Weather data
- Effective means for proof of authorship.
  - Signature and data are the same object.
- Effective means of tamper proofing.
  - Integrity information is embedded in the data.

Why is Watermarking Possible ?

- Real-world datasets can tolerate a small amount of error without degrading their usability
  - Meteorological data used in building weather prediction models, the wind vector and temperature accuracies in this data are estimated to be within 1.8 m/s and 0.5 ºC.
  - Such constraints bound the amount of change or alteration to that can be performed on the data.

What defines the usability constraints ?

- Usability constraints are application dependent.
  - Alterations performed by the watermark embedding should be unidentifiable by the human visual system in images/video.
  - For consumer behavior data: watermarking should preserve periodicity properties of the data.

What defines the usability constraints ? (Cont.)

Courtesy of http://maps.google.com

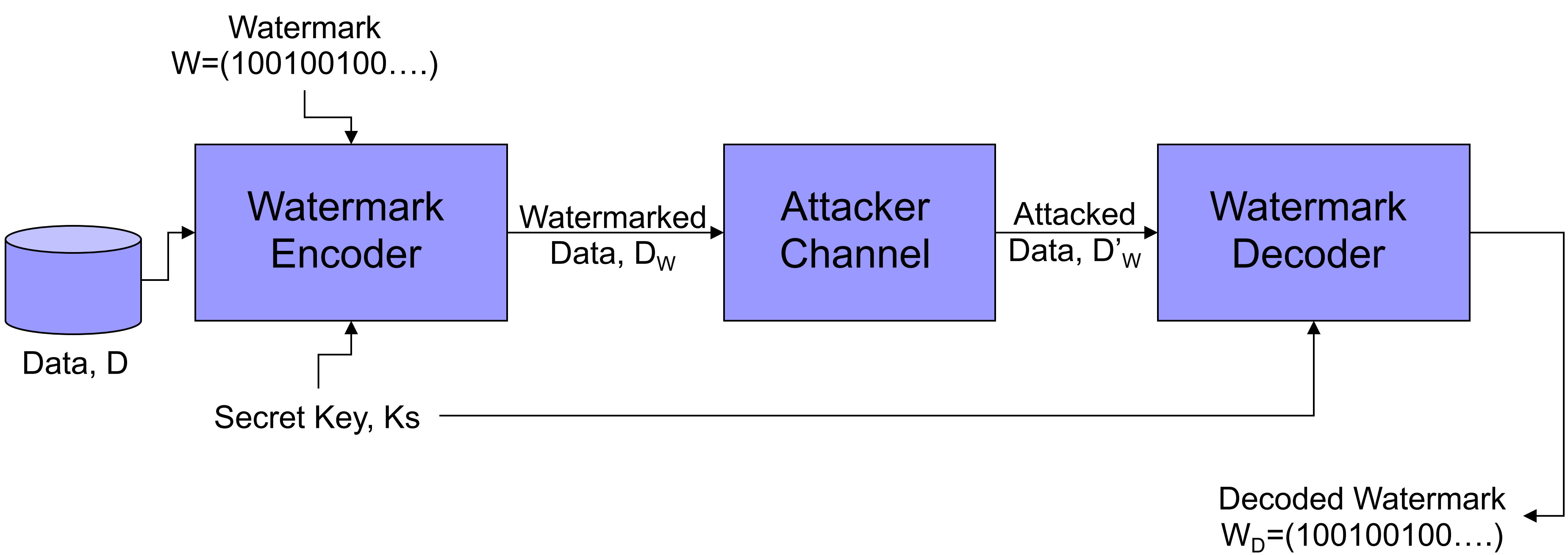
Watermark Desirable Properties

- Detectability (Key-Based System)
  - Can be easily detected only with the knowledge of the secret key.
- Robustness
  - Watermark cannot be easily destroyed by modifying the watermarked data.
- Imperceptibility
  - Presence of the watermark is unnoticeable.
- Blind System
  - Watermark detection does not require the knowledge of the original data.

Outline

- Introductory Material
- General Watermarking Model & Attacks
- WM Technique 1 (Agrawal et al.)
- WM Technique 2 (Sion et al.)
- Future Challenges and References

Watermarking Model



Relational and multimedia data

- A multimedia object consists of a large number of bits, with considerable redundancy. Thus, the large watermark hiding bandwidth.
- The relative spatial/temporal positioning of various pieces of a multimedia object typically does not change. Tuples of a relation on the other hand constitute a set and there is no implied ordering between them.
- Portions of a multimedia object cannot be dropped or replaced arbitrarily without causing perceptual changes in the object. However, a pirate of a relation can simply drop some tuples or substitute them with tuples from other relations.

Attacker Model

- Attacker has access to only the watermarked data set.
- The attacker's goal is to weaken or even erase the embedded watermark and at the same time keep the data usable. "Attacker's Dilemma"
- Possible Attacks
  - Tuple deletion
  - Tuple alteration
  - Tuple insertion

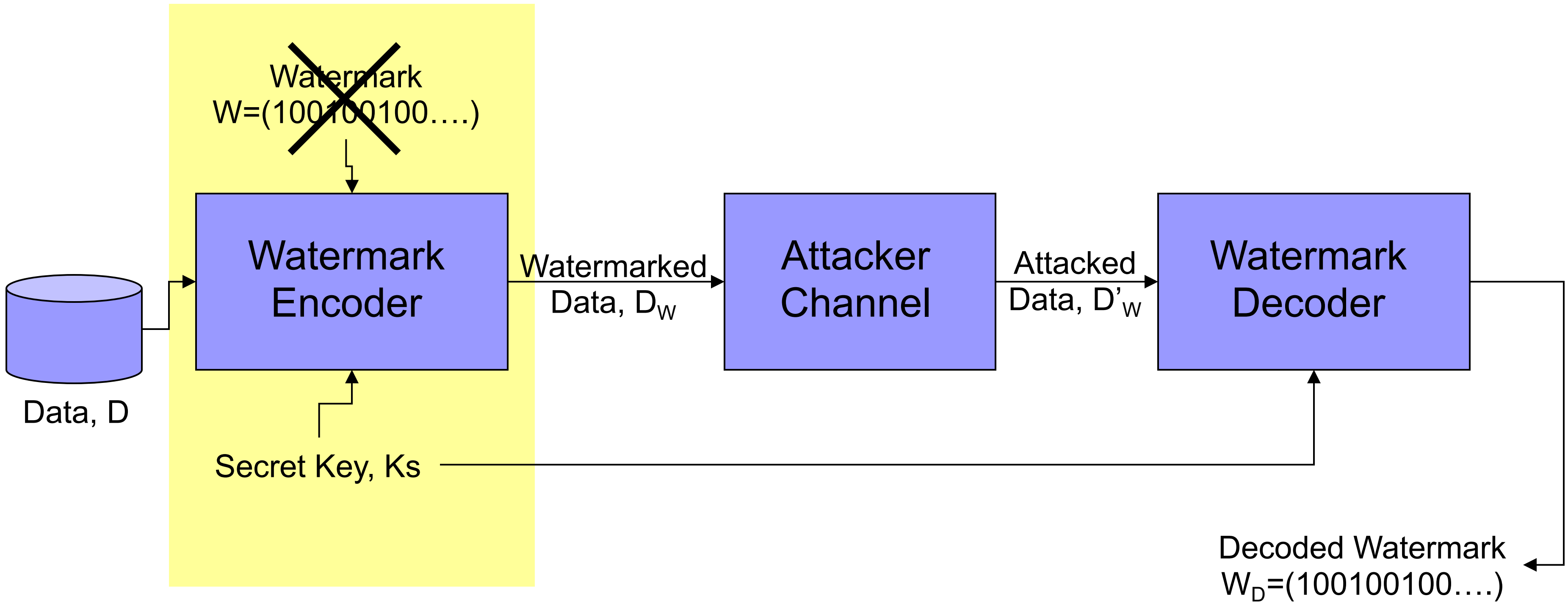
Outline

- Introductory Material
- General Watermarking Model & Attacks
- WM Technique 1 (Agrawal et al.)
- WM Technique 2 (Sion et al.)
- Future Challenges and References

WM Technique 1 (Agrawal et. al.)

- Watermarking of numerical data.
- Technique dependent on a secret key.
- Uses markers to locate tuples to hide watermark bits.
- Hides watermark bits in the **least significant bits.**

WM Technique 1: Encoder



Instead:

Watermark is a function of the data and the secret key

WM Technique 1: Encoder

- Assumptions
  - *K*, *e*, *m* and *v* are selected by the data owner and are kept secret.
  - *"K" is the secret key.*
  - *"e"* least significant bits can be altered in a number without affecting its usability. Example, *e=3, 101101101.1011101*
  - *"m"* used for marker selection and 1/m is fraction of tuples marked
  - *"v"* is the number of attributes used in the watermarking process.

Message Authentication Code

- *One way hash function H operates on an input message M of arbitrary length and returns a fixed length of has value h.*
- *Three characteristics*
  - *Given M, it is easy to compute h*
  - *Given h, it is hard to compute M*
  - *Give M, it is hard to find another message M' such that H(M) = H(M')*
- *A message authentication code (MAC) is a one-way has function that depends on a key.*

  MAC(r.P) = MAC(r.P) = H(K || MAC(K||r.P)
- *r.P is the primary key attribute of relation r, K is a secret key known only to owner, and output is an integer value in a wide range.*

WM Technique 1: Encoder

- *For all tuples r in D*
  - *MAC(r.P) = MAC(r.P) = H(K || MAC(K||r.P)*
  - *if(MAC(r.P) mod m == 0)* *// Marker Selection*
    - *i = (MAC(r.P) mod v* *// Selected Attribute*
    - *b = (MAC(r.P) mod e* *// Selected LSB index*
    - *if((MAC(r.P) mod 2 == 0)* *// MAC is even*
      - *Set bit b of r.A_i*
    - *Else*
      - *Clear bit b of r.A_i*

WM Technique 1 : Encoder

(MAC mod v=1

MAC mod m
1
4
0
9

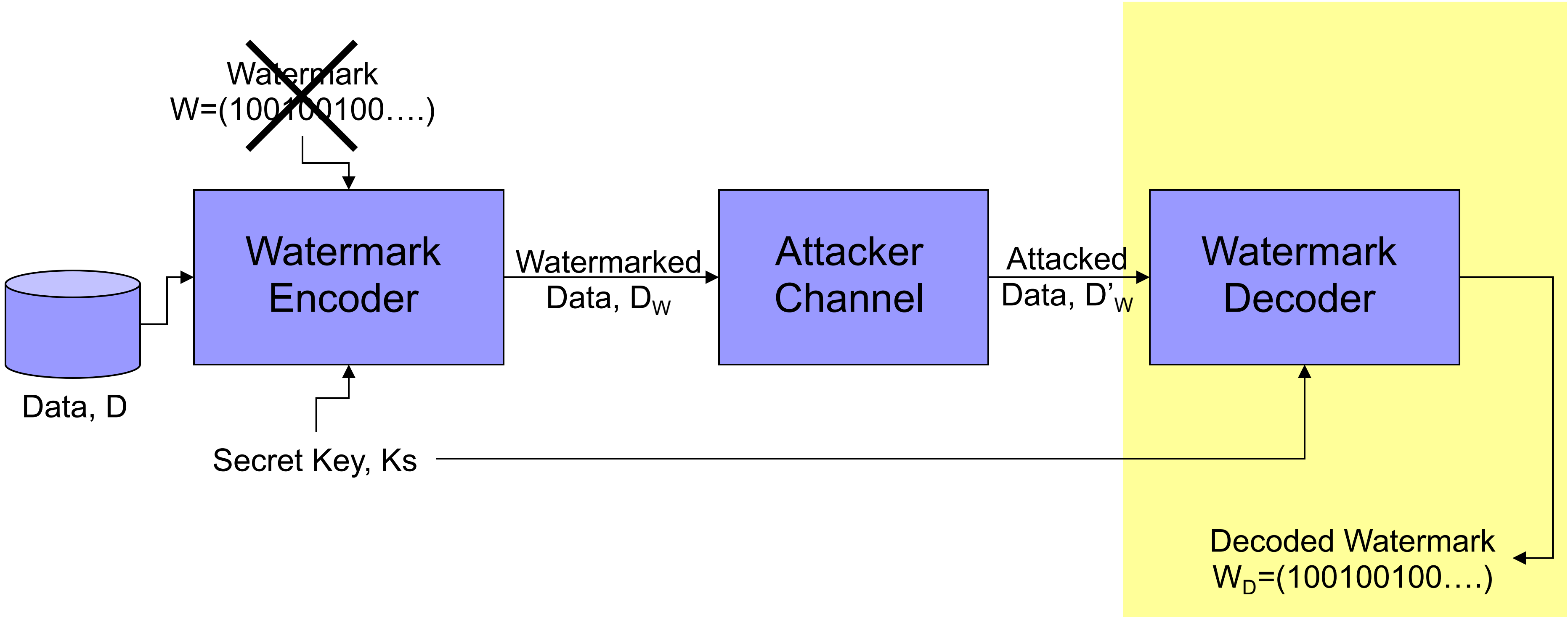
PKey	*Attribute 0*	*Attribute 1*	……….	*Attribute v-1*
1234				
2345				
3390				
4455				

1010101010.010111011

MAC mod e

MAC is MAC(K || MAC(K || r.P))

WM Technique 1 : Decoder



WM Technique 1 : Decoder

- *Match = Total_Count = 0*
- *For all tuples r in D*
  - *r.MAC = H(K||r.P||K)*
  - *if(r.MAC mod m == 0)   // Marker Selection*
    - *Total_Count++*
    - *i = r.MAC mod v   // Selected Attribute*
    - *b = r.MAC mod e   // Selected LSB index*
    - *if(r.MAC mod 2 == 0) // MAC is even*
      - *if bit b of r.A_i is Set*
        - *Match++*
    - *Else*
      - *If bit b of r.A_i is Clear*
        - *Match++*
- *Compare (Match/Total_count) > Threshold*

WM Technique 1 : Decoder

MAC mod v → Attribute 1

MAC mod m
1
4
0
9

PKey	*Attribute 0*	*Attribute 1*	……….	*Attribute v-1*
1234				
2345				
3390				
4455				

1010101010.010111011

↑ *MAC mod e*

MAC is MAC(K || MAC (K || r.P))

WM Technique 1 : Strengths

- Computationally efficient *O(n)*
  - Tuple sorting not required.

- Incremental Updatability

WM Technique 1 : Weaknesses

- No provision of multi-bit watermark, all operations are dependent only on the secret key.
- Not resilient to alteration attacks. Least Significant Bit (LSB) can be easily manipulated by simple numerical alterations
  - Shift LSB bits to the right/left.
- Requires the presence of a primary key in the watermarked relation.
- Does not handle other usability constraints such as:
  - Category preserving usability constraints.

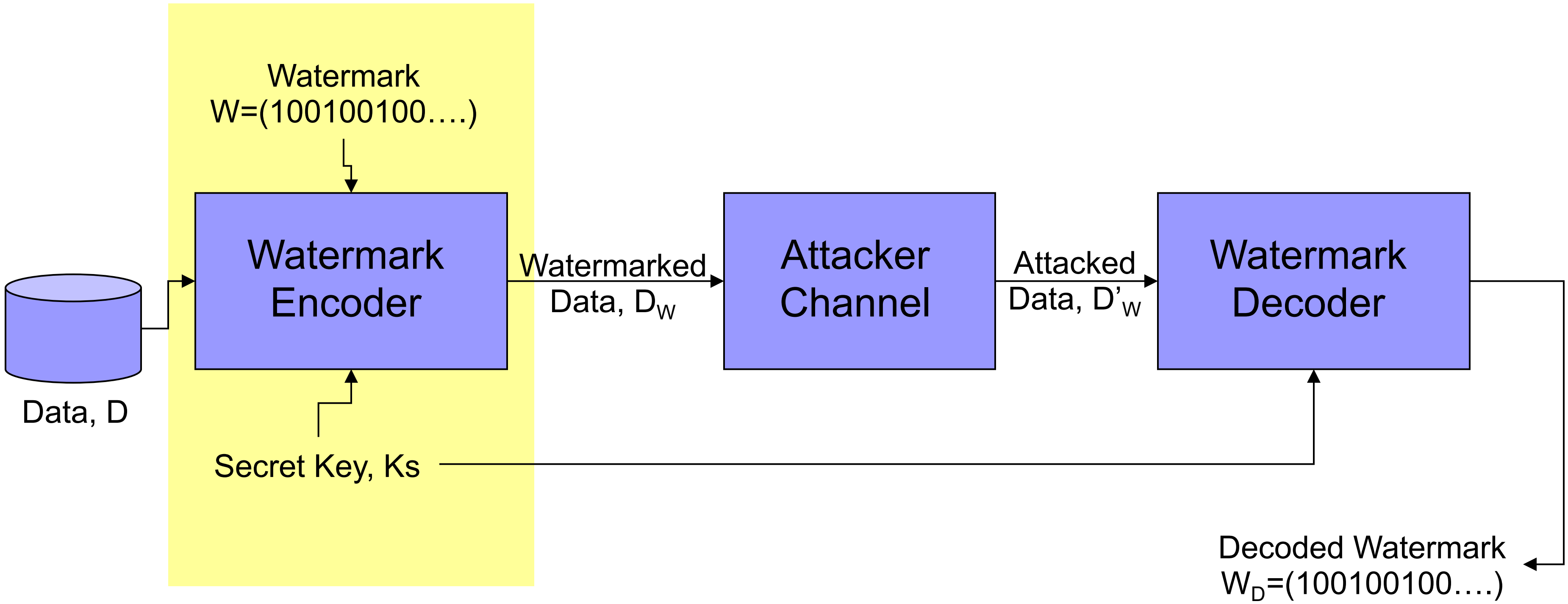
Outline

- Introductory Material
- General Watermarking Model & Attacks
- WM Technique 1 (Agrawal et al.)
- WM Technique 2 (Sion et al.)
- Future Challenges and References

WM Technique 2 :(Sion et. al.)

- Watermarking of numerical data.
- Technique dependent on a secret key.
- Instead of primary key uses the most significant bits of the *normalized* data set.
- Divides the data set into partitions using markers.
- Varies the partition statistics to hide watermark bits.

WM Technique 2 : Encoder

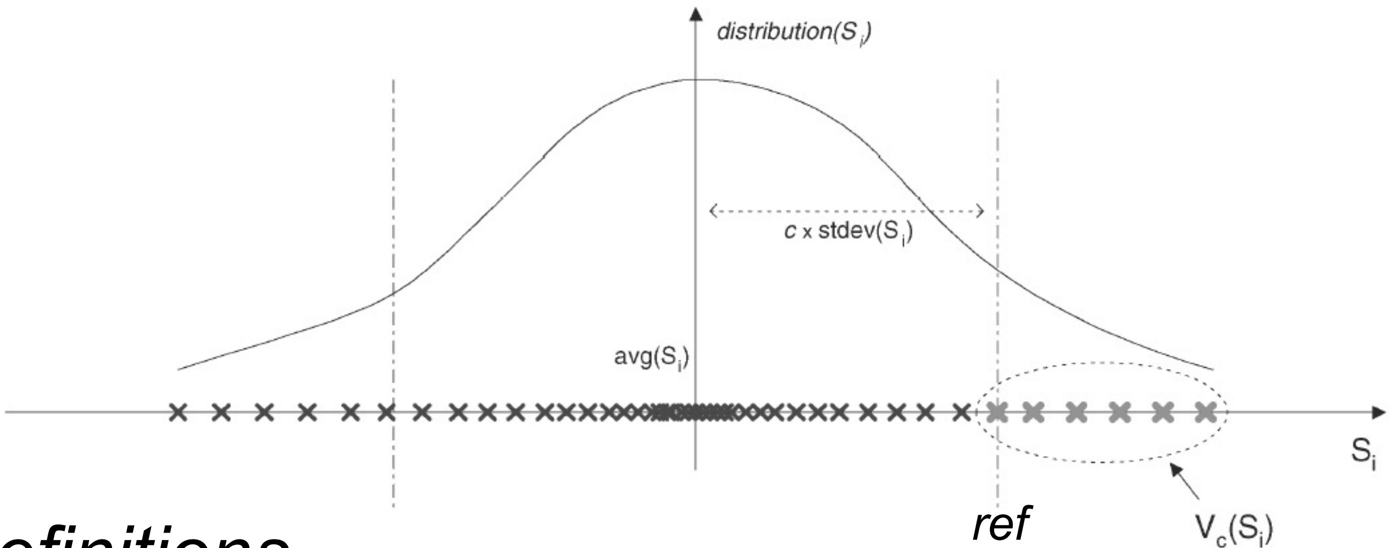


WM Technique 2: How to hide a single bit in a number set ?

- Problem:

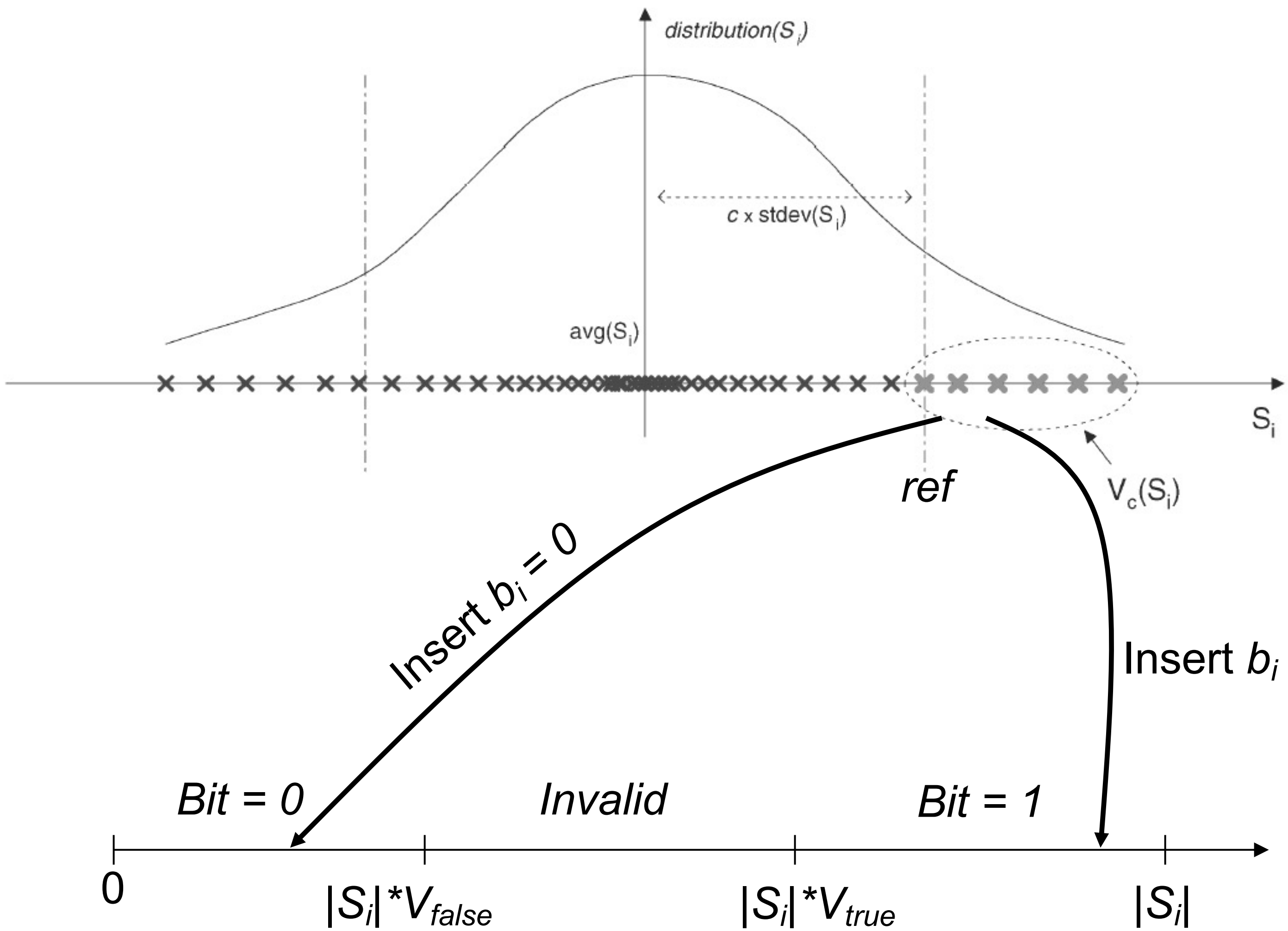
  " Given a number set $S_i = \{s_1, \ldots, s_n\}$, *how to vary their statistics to embed bit* b_i*. Subject to the provided usability constraints."*

Paper 2: How to hide a single bit in a number set ?



- *Definitions*
  - μ = *mean(S_i)*
  - σ = *stdev(S_i).*
  - ***ref =*** μ ***+ c***σ***, c is a confidence factor***
  - *Vc(S_i) = number of points greater than **ref**. We refer to them as "**positive violators**".*

Paper 2: How to hide a single bit in a number set ?



WM Technique 2: How to avoid using the primary key ?

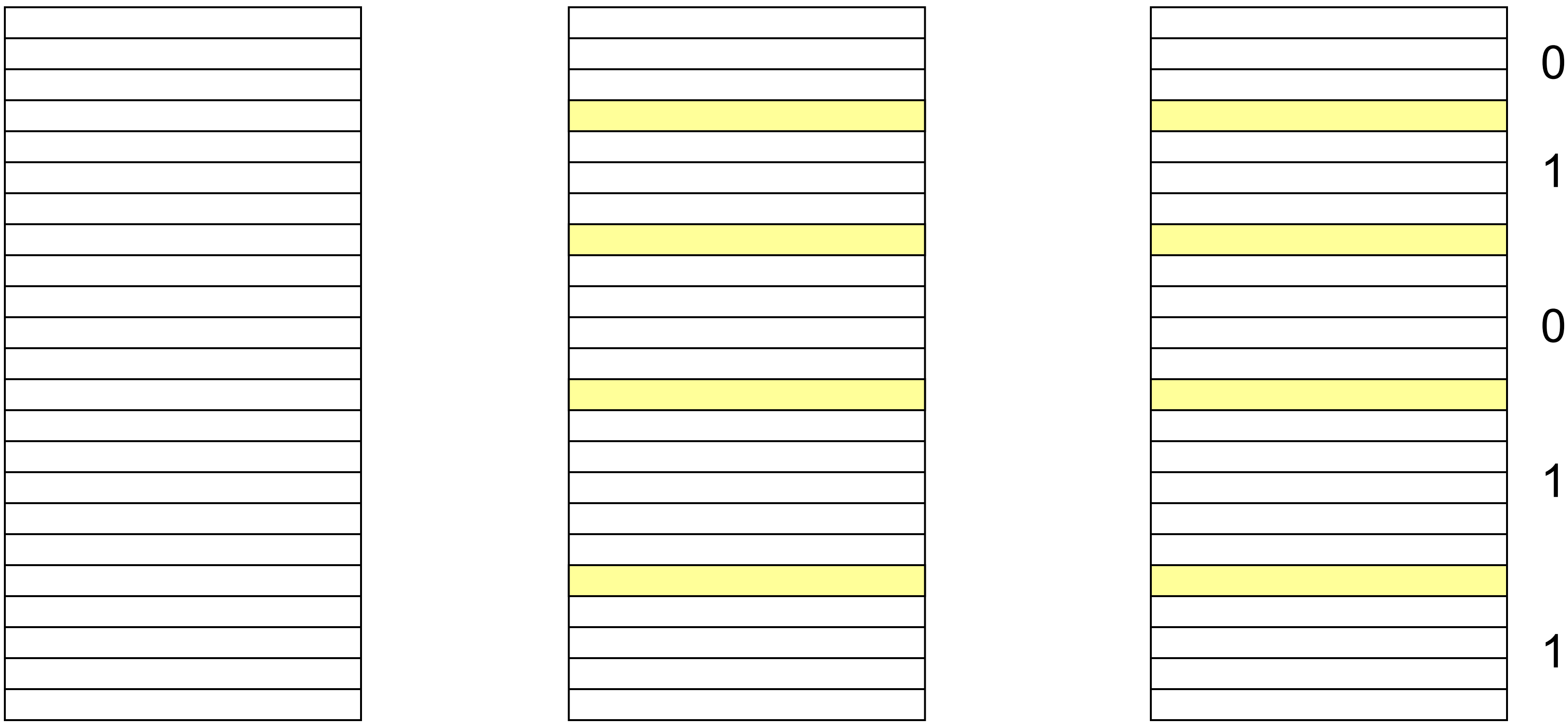
- Given a number set $S_i = \{s_1, \ldots, s_n\}$, generate $Norm(Si) = S_i / max(S_i)$.

- For each number in s_k in $Norm(Si)$ use the first *n* most significant bits (MSB) as the primary key for s_k.

WM Technique 2 : Encoder

- Step 1: (Sorting)
  - Compute the MAC of each tuple:
    - *r.MAC = H(K || r.P || K)* *// r.P = MSB(r.A)*
  - Sort tuples in ascending order using the computed MAC.
- Step 2: (Partitioning)
  - Locate markers: tuples with *r.MAC mod m = 0*
  - Tuples between two markers are in the same partition.
- Step 3: (Bit Embedding):
  - Embed a watermark bit in each partition using the bit embedding technique discussed earlier.

WM Technique 2 : Encoder

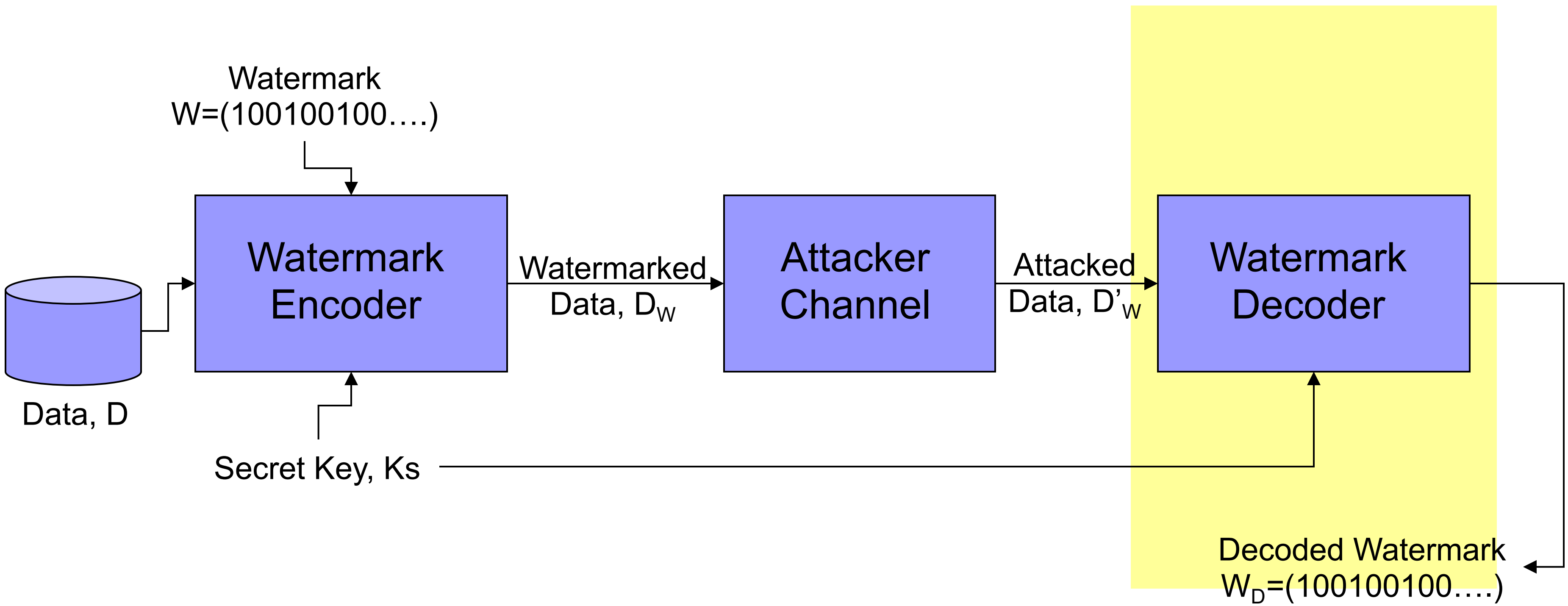


Step 1
Sort Ascending
According to MAC

Step 2
Locate Markers
r.MAC mod m = 0

Step 3
Bit Embedding

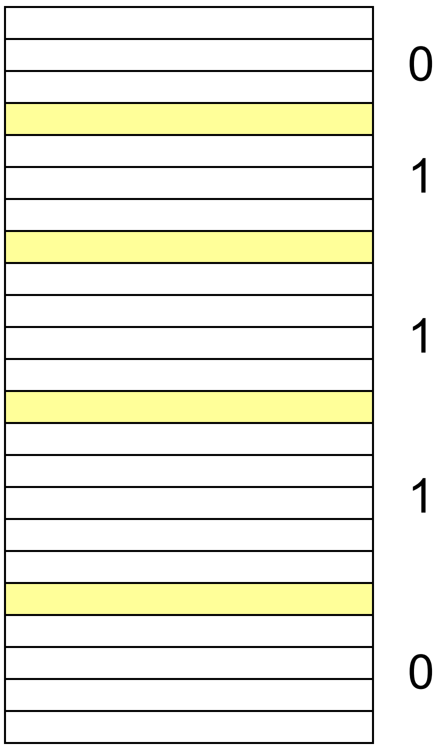
WM Technique 2 : Decoder



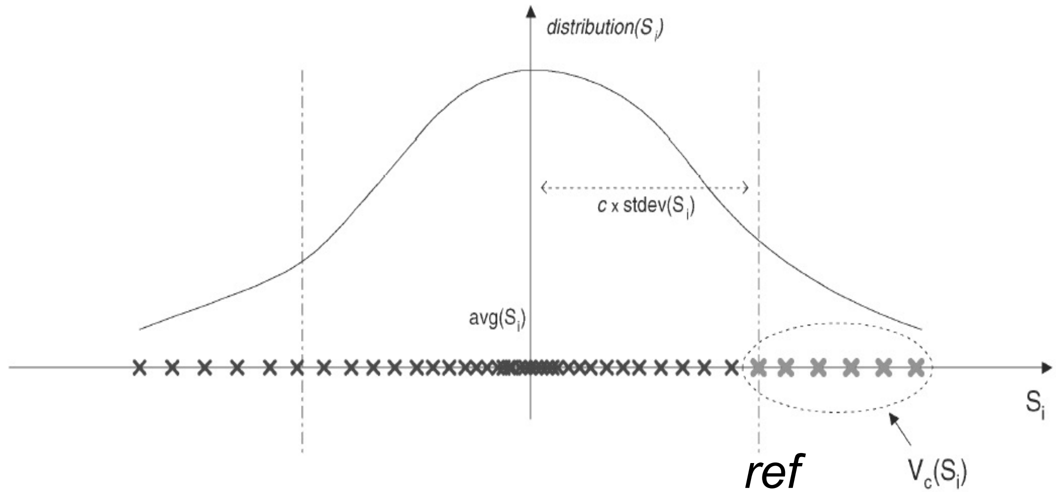
WM Technique 2 : Decoder

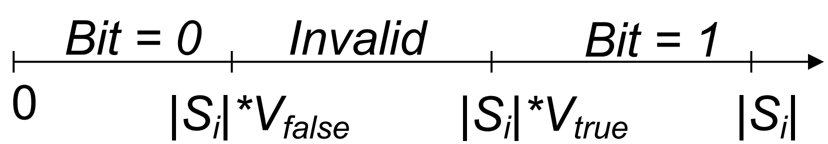
- Step 1: (Sorting & Partitioning)
  - Partition data set using the same approach used in the encoding phase.
- Step 2: (Bit Detection)
  - For each partition S_i compute $V_c(S_i)$ and decode the embedded bit.
- Step 3: (Majority Voting):
  - Watermark bits are embedded in several partitions use majority voting to correct for errors.

WM Technique 2 : Decoder



Watermarked Data Set





bits	5	4	3	2	1	0
w_0	1	0	1	1	1	0
w_1	1	0	1	0	1	0
w_2	1	0	0	0	1	1
w_{result}	**1**	**0**	**1**	**0**	**1**	**0**

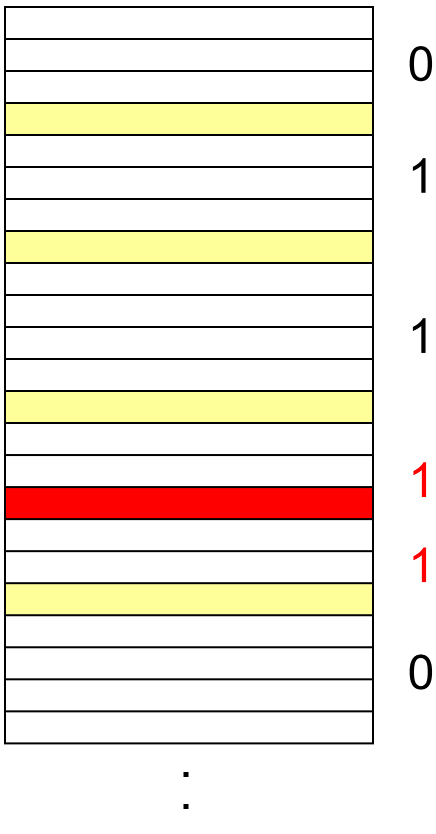
Majority Voting

WM Technique 2 : Strengths

- Bit embedding technique honors usability constraints.

- Embeds watermark in data statistics which makes technique more resilient to alteration attacks compared with Least Significant Bits (LSB).

WM Technique 2 : Watermark Synchronization Error (Tuple Addition)

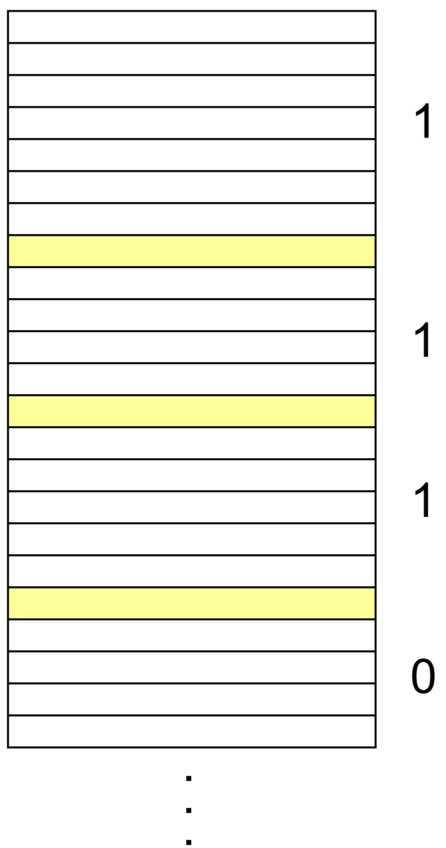


Watermarked Data Set

	5	4	3	2	1	0
W_0	1	0	1	1	1	0
W_1	1	0	1	0	1	0
W_2	1	0	0	0	1	1
W_{result}	**1**	**0**	**1**	**0**	**1**	**0**

	5	4	3	2	1	0
W_0	0	1	1	1	1	0
W_1	0	1	0	1	0	1
W_2	0	0	0	1	1	1
W_{result}	**0**	**1**	**0**	**1**	**1**	**1**

WM Technique 2 : Watermark Synchronization Error (Tuple Deletion)



Watermarked Data Set

	5	4	3	2	1	0
W_0	1	0	1	1	1	0
W_1	1	0	1	0	1	0
W_2	1	0	0	0	1	1
W_{result}	**1**	**0**	**1**	**0**	**1**	**0**

W_0	0	1	0	1	1	1
W_1	1	1	0	1	0	1
W_2	x	1	0	0	0	1
W_{result}	**x**	**1**	**0**	**1**	**0**	**1**

Paper 2: Weaknesses

- Watermark suffers badly from watermark synchronization error cause by
  - Tuple deletion attacks.
  - Tuple addition attacks.
- No optimality criteria when choosing the decoding thresholds
  - Errors even in absence of attacker.
- No clear systematic approach for manipulating data
  - Only a very small space of the feasible data manipulations investigated.

Outline

- Introductory Material
- General Watermarking Model & Attacks
- WM Technique 1 (Agrawal et al.)
- WM Technique 2 (Sion et al.)
- Future Challenges and References

Challenges

- Investigate watermarking other types of data. Such as data streams.
- Design robust watermarking techniques that are resilient to watermark synchronization errors.
- Design a fragile watermarking technique for relational databases.

References

- J. Kiernan, R. Agrawal, "Watermarking Relational Databases," *Proc. 28th Int'l Conf. Very Large Databases VLDB*, 2002.

- Radu Sion, Mikhail Atallah, Sunil Prabhakar, "Rights Protection for Relational Data," *IEEE Transactions on Knowledge and Data Engineering*, Volume 16, Number 6, June 2004

Questions?