Don’t Abandon the Primary Key: A High-Synchronization and Robust Virtual Primary Key Scheme for Watermarking Relational Databases

Ke Yang[1,2,3] Shuguang Yuan[1,2,3] Jing Yu[1,2,3] Yuyang Wang[1,2,3] Tengfei Yang[4] Chi Chen[1,2,3]

[1] Institute of Information Engineering, Chinese Academy of Sciences

[2] School of Cyber Security, University of Chinese Academy of Sciences

[3] Key Laboratory of Cyberspace Security Defense

[4] National Computer Network Emergency Response Technical Team/Coordination Center of China

Background

- Relational databases are still the data storage and sharing solution in most enterprises and applications.
- The demand for databases publishing and sharing has increased significantly.
- Copyright issues become a significant concern.
  - Unauthorized database distribution/modification
  - Database piracy

Background

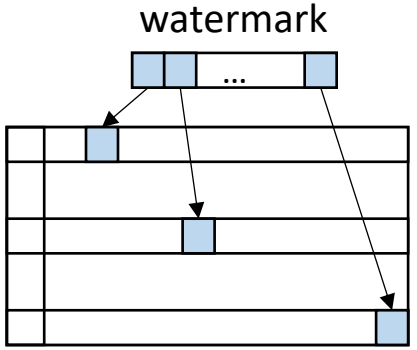
- Relational databases are still the data storage and sharing solution in most enterprises and applications.
- The demand for databases publishing and sharing has increased significantly.
- Copyright issues become a significant concern.
  - Unauthorized database distribution/modification
  - Database piracy
- **Database Watermarking techniques** can effectively protect copyright.

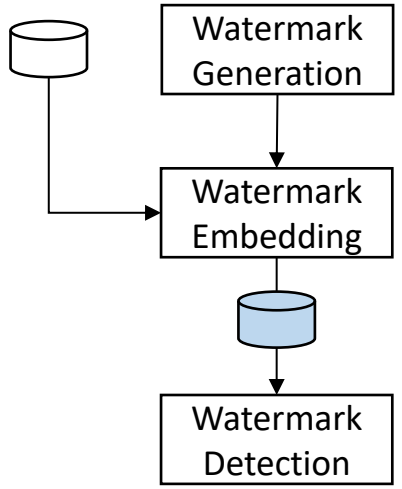
Background

What is database watermarking and how it work?

- Database watermarking secretly embeds hidden markers or watermark information within a database to protect copyright.

- Workflow of database watermarking:
  - Watermark Generation
  - Watermark Embedding
  - Watermark Detection

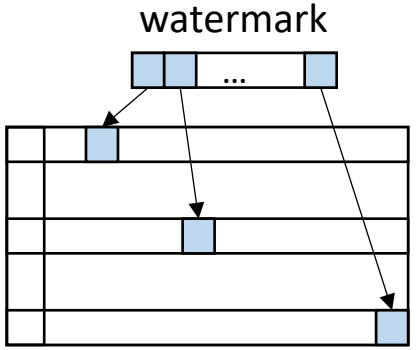




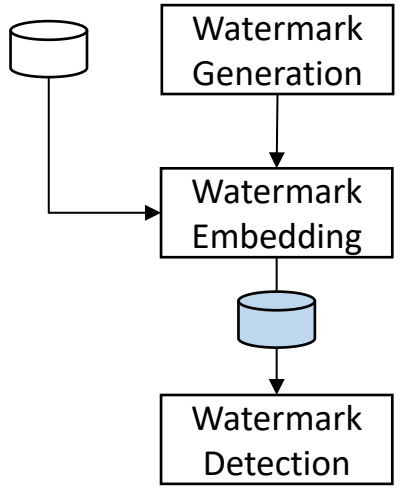
Background

What is database watermarking and how it work?

- Database watermarking secretly embeds hidden markers or watermark information within a database to protect copyright.



- Workflow of database watermarking:
  - Watermark Generation
  - Watermark Embedding
  - Watermark Detection



- Main goals: Prevent copyright issues, and provide evidence of databases ownership by **detecting watermarks** in case of disputes.

Background

Watermark synchronization

The process of aligning the detected watermark with the embedded watermark.

PK scheme

- Take advantage of the uniqueness of primary key
- Achieve high watermark synchronization

Vulnerability

- Primary key attacks: Cannot detect watermark when primary key is erased or changed.

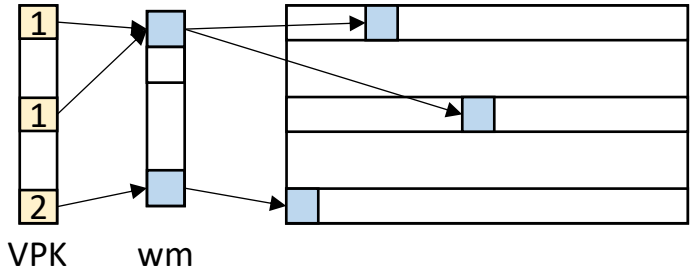
Background

VPK scheme

- Generate virtual primary key for watermarking instead of primary key
- Address primary key attacks

Vulnerability

- Synchronization problems: The redundant values of virtual primary keys affect the embedding quality and watermark synchronization.
- Fail to utilize primary key: Even primary key remains unchanged, which decrease detection ability in many cases.



Background

Robust against attacks

- **Subset attacks** (distortion of tuples)
- **Attribute attacks** (distortion of attribute)
- **Additive attacks** (re-watermarking)

Background

Robust against attacks

- **Subset attacks** (distortion of tuples)
- **Attribute attacks** (distortion of attribute)
- **Additive attacks** (re-watermarking)

New risk we found

- **Attribute name attacks:** Attacks on attribute names (distortion of attribute names)
- Attribute name attacks can compromise watermark detection by **distorting the link** between detected and watermarked attributes.

Research Objectives & Contributions

Objectives

- Resist attribute name attacks
- Avoid primary key attacks & Mitigate synchronization problems
- Utilize primary key if it is intact
- Evaluation

Contributions

- A attribute classifier to maintain the correctness of attributes
- A high-synchronization and robust VPK scheme
- Watermarking algorithm integrates primary key
- The performance is verified by experiments

Research Objectives & Contributions

Objectives

- Resist attribute name attacks
- Avoid primary key attacks & Mitigate synchronization problems
- Utilize primary key if it is intact
- Evaluation

Contributions

- A attribute classifier to maintain the correctness of attributes
- A high-synchronization and robust VPK scheme
- Watermarking algorithm integrates primary key
- The performance is verified by experiments

PK	A_1	A_2	A_3	A_4	A_5
001	43	193	2	**710**	433
002	**15**	358	3	549	487
003	21	670	8	967	**404**
004	39	**454**	2	828	469

(a) PK scheme

PK	A_1	A_2	A_3	A_4	A_5
001	43	193	2	710	433
002	17	358	3	549	**486**
003	21	**672**	8	967	404
004	**39**	455	2	828	469

(b) VPK scheme

A_1 A_2 A_3 A_4 A_5

Classifier

PK	L_1	L_2	L_3	L_4	L_5
001	43	**192**	2	710	**432**
002	17	358	3	549	487
003	21	**671**	**9**	967	404
004	39	455	2	828	469

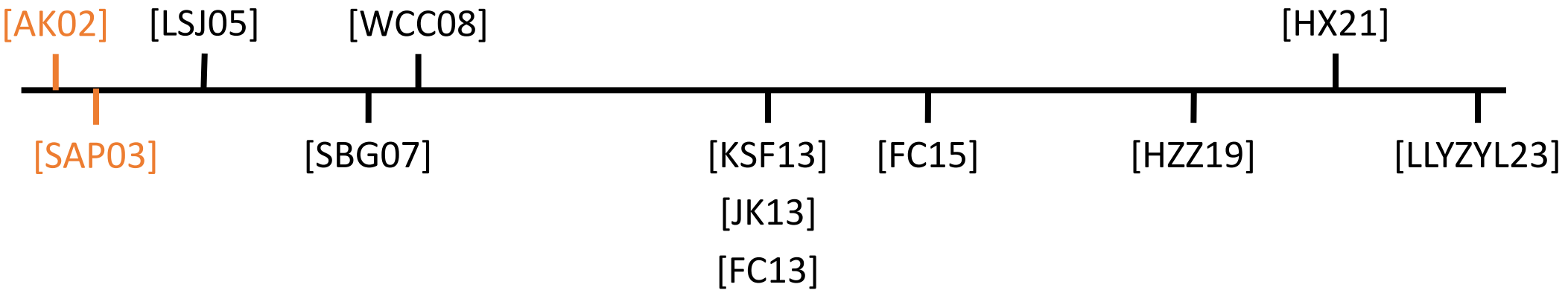
(c) Our scheme

Related Works

PK schemes

Bit-level Watermarking

Robust, meaningless watermark → Fingerprint, meaningful watermark, reversibility



Statistics-based → Distortion control: Optimization, DE, HS, semantic-preserved

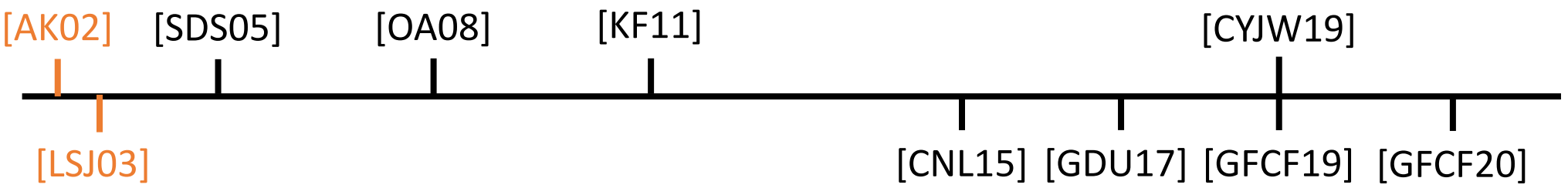
Statistics/semantic-based Watermarking

Related Works

VPK schemes

Non-primary key attributes as virtual primary key

Focus on statistics preserving and alternative attributes selecting



Focus on non-numerical relations and mitigate synchronization problems

Pseudo-random selected values in tuples to generate virtual primary key

Related Works

Attacks definition

- [KF18]
  - Insertion attacks, Deletion attacks, Alteration attacks
  - Multifaceted attacks
  - Additive (Re-watermarking) attacks

- [KSY20]
  - Substitution, addition, alteration
  - Vertical partition
  - Mix-Match

- [YCYYY22]
  - Subset attacks (Insertion attacks, Deletion attacks, Alteration attacks)
  - Attribute attacks

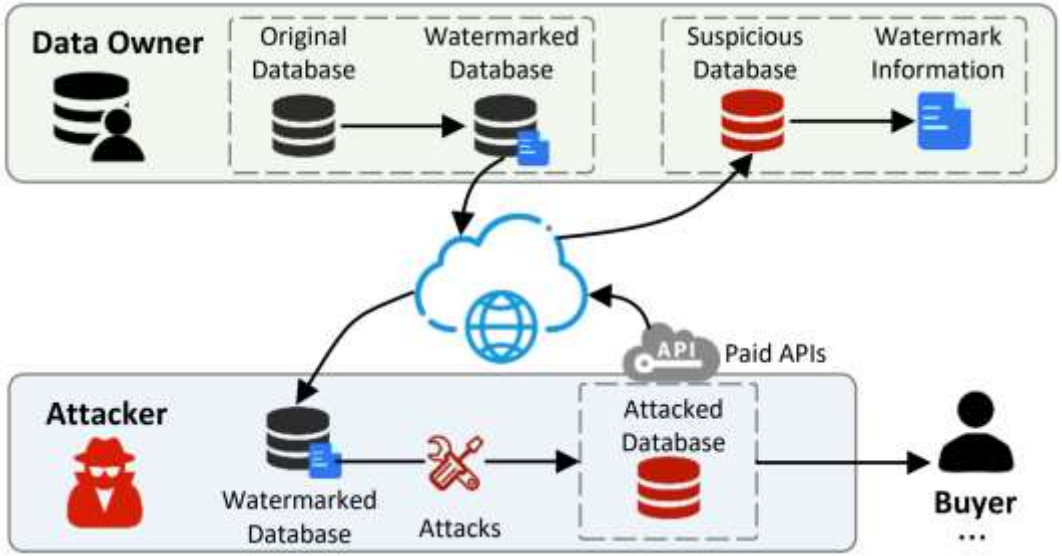
Threat Model

Defender

- Assumption: Watermark database + Detect watermark from suspicious database.

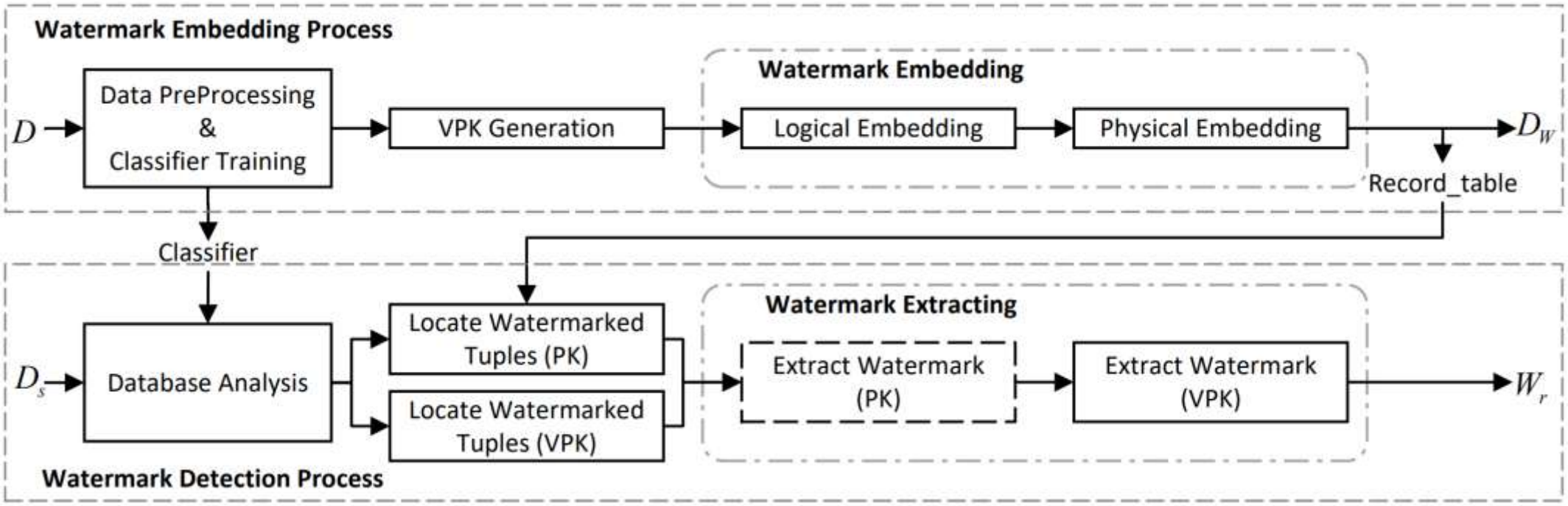
Attacker

- Goals: Bypass watermark detection + Maintain required data quality
- Capabilities: Access watermarking scheme + Alter database by attacks



Approach Overview

Workflow of watermarking scheme



Attribute Attacks Extension

Attribute column attacks

- Traditional attribute attacks: column deletion, insertion, or shuffling

Attribute Attacks Extension

Attribute column attacks

- Traditional attribute attacks: column deletion, insertion, or shuffling

Attribute name attacks

- Attribute name shuffling, deletion and substitution
- Compromise attribute order
- Only [LSJ03] and [GFCF19] can resist

Attribute Attacks Extension

Attribute column attacks

- Traditional attribute attacks: column deletion, insertion, or shuffling

Attribute name attacks

- Attribute name shuffling, deletion and substitution
- Compromise attribute order
- Only [LSJ03] and [GFCF19] can resist

Mix-match attribute attacks

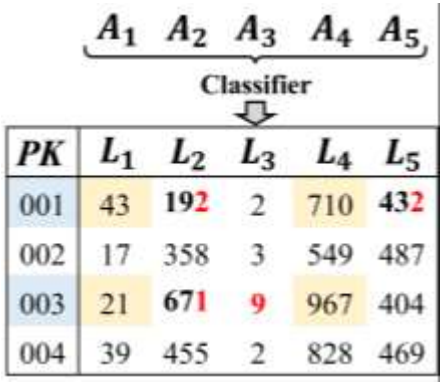
- Combinations of attribute column attacks and attribute name attacks
- Compromise attribute order & count

Data Preprocessing and Classifier Training

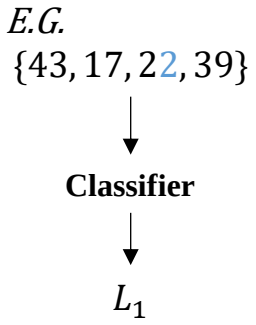
Q: How to resist Attribute name attacks and Mix-match attribute attacks?

Labeling attribute names and **train a classifier** to identify attribute labels.

Features: minimum, maximum, mean, variance, skewness, kurtosis, entropy of attributes.



PK	L_1	L_2	L_3	L_4	L_5
001	43	192	2	710	432
002	17	358	3	549	487
003	21	671	9	967	404
004	39	455	2	828	469

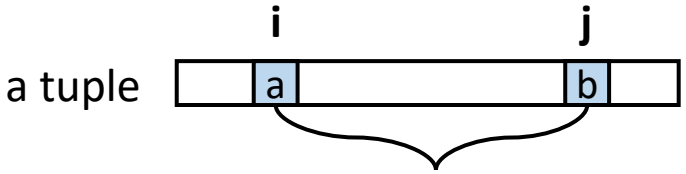


Accuracy: 96%, τ on prediction probabilities to trade off FP and FN.

Virtual Priamry Key Generation

Candidate Attributes with Labels

- Selected to generate virtual primary key
- Decrease duplicate virtual primary key



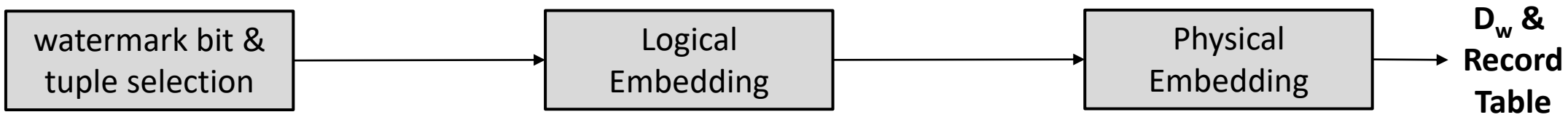
vpk: Hash Combination [H(a·i), H(b·j), H(a·i b·j)]

vpk_dict: a·i → H(a·i), H(a·i b·j) b·j → H(b·j), H(a·i b·j)

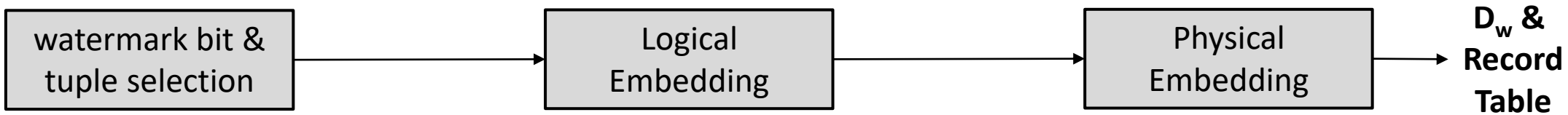
A Candidate Attribute value map to multiple virtual primary keys

- Increase watermark capacity & Mitigate synchronization problems
- Resist attribute column attacks

Watermark Embedding

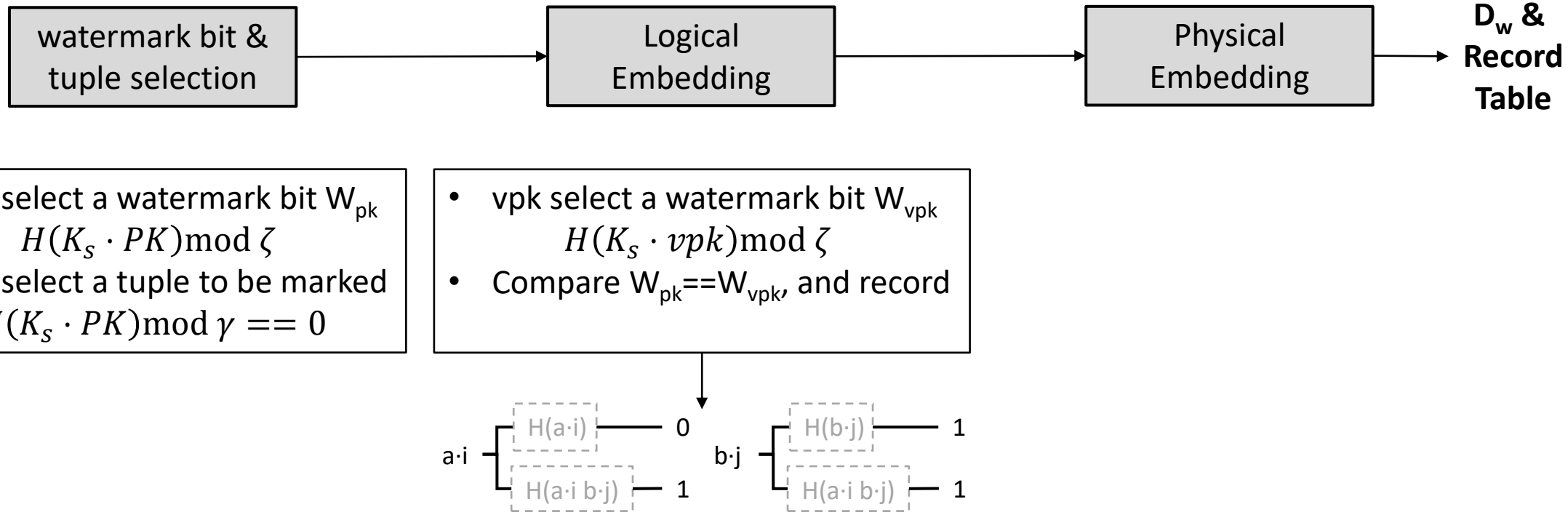


Watermark Embedding

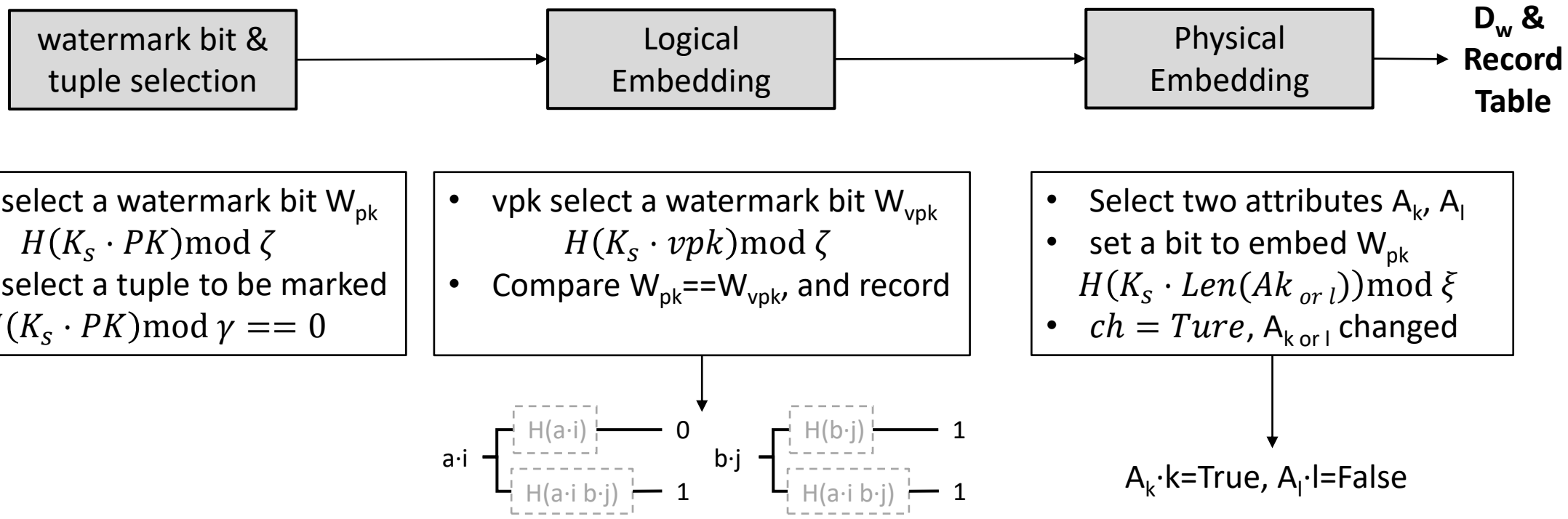


- PK select a watermark bit W_{pk}
  $H(K_s \cdot PK) \bmod \zeta$
- PK select a tuple to be marked
  $H(K_s \cdot PK) \bmod \gamma == 0$

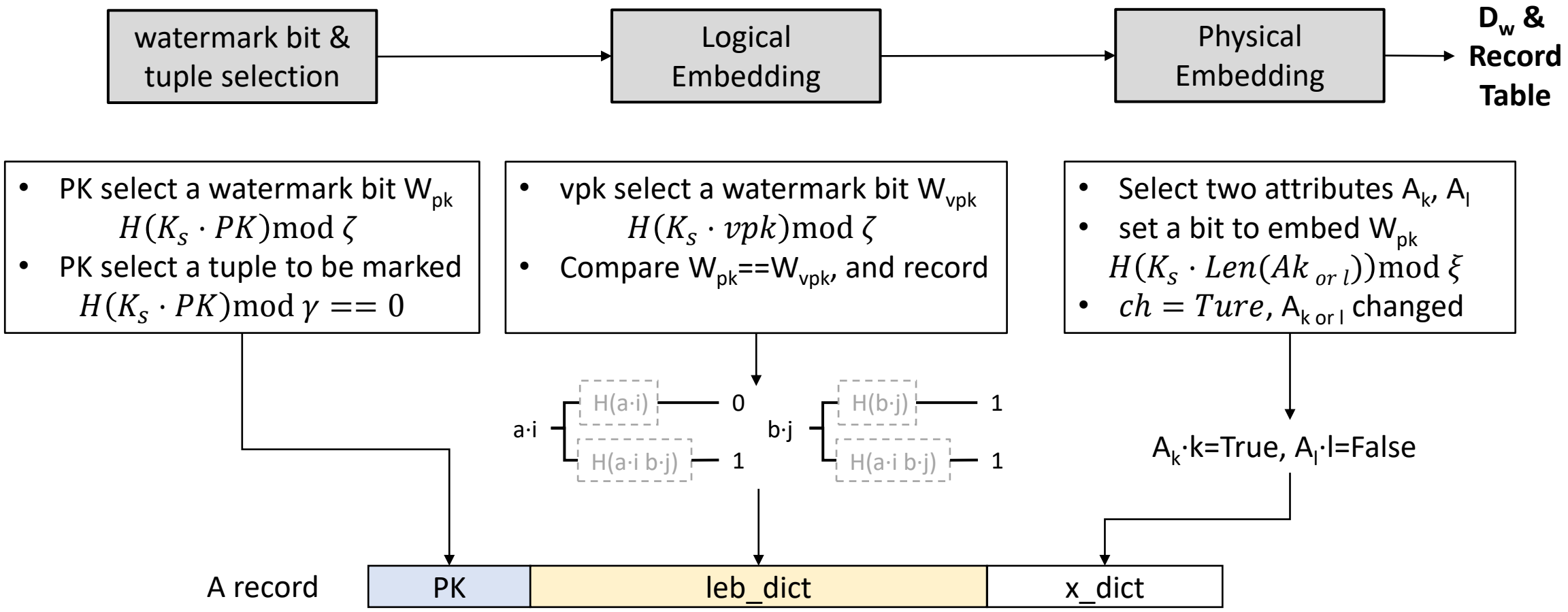
Watermark Embedding



Watermark Embedding



Watermark Embedding



Watermark Detection

Database Analysis

- Attack analysis, damage analysis
- Analysis of whether the label needs to be restored by attribute classifier

Watermark Detection

Database Analysis

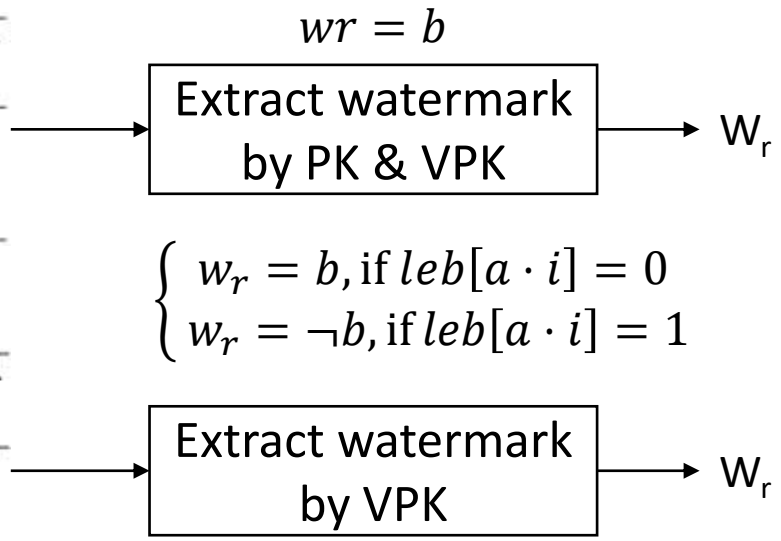
- Attack analysis, damage analysis
- Analysis of whether the label needs to be restored by attribute classifier

Match record when Primary Key is Available

Matched PK	A_{w_i} of x_dict	$t.A_{w_i}$ of x_dict	$t.A_{w_i}$ of tuple t	Located water-marked tuple
9	A_{w_0}, A_{w_1}	7,3	7,3	True
12	A_{w_0}, A_{w_1}	24,5	22,7	False
37	A_{w_0}, A_{w_1}	13,2	13,3	True

Match record when Primary Key is Unavailable

$\Phi_{C'}$	Φ_C of leb_dict	A_{w_i} of x_dict	$t.A_{w_i}$ of x_dict	$t.A_{w_i}$ of tuple t	Located water-marked tuple
$234 \circ C_i$	$234 \circ C_i, 32 \circ C_j$	A_{w_0}, A_{w_1}	7,3	7,3	True
$256 \circ C_i$	$256 \circ C_i, 54 \circ C_j$	A_{w_0}, A_{w_1}	24,5	22,7	False
$197 \circ C_i$	$197 \circ C_i, 69 \circ C_j$	A_{w_0}, A_{w_1}	13,2	13,3	False



Experiments

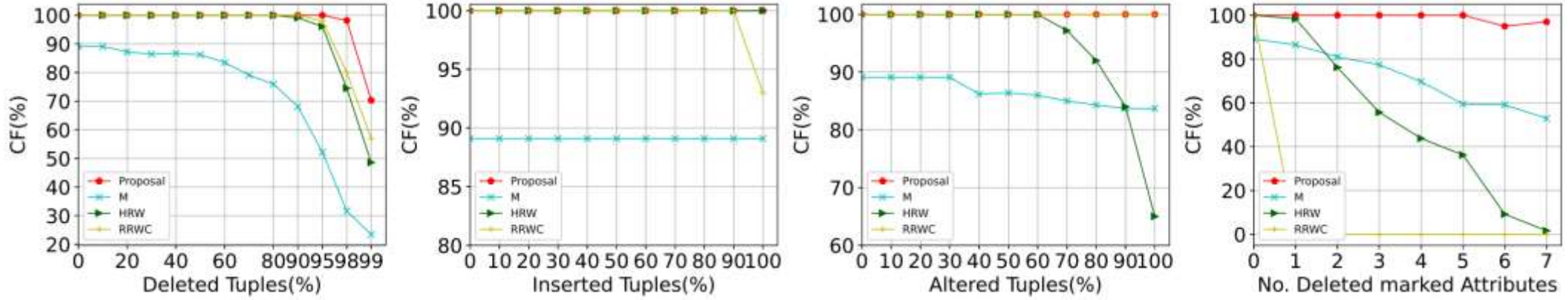
Dataset: Forest Cover Type (581012 tuples)
Control: around 2900 watermarked tuples

Robustness Verification:
VPK watermarking algorithm

Synchronization problems mitigation

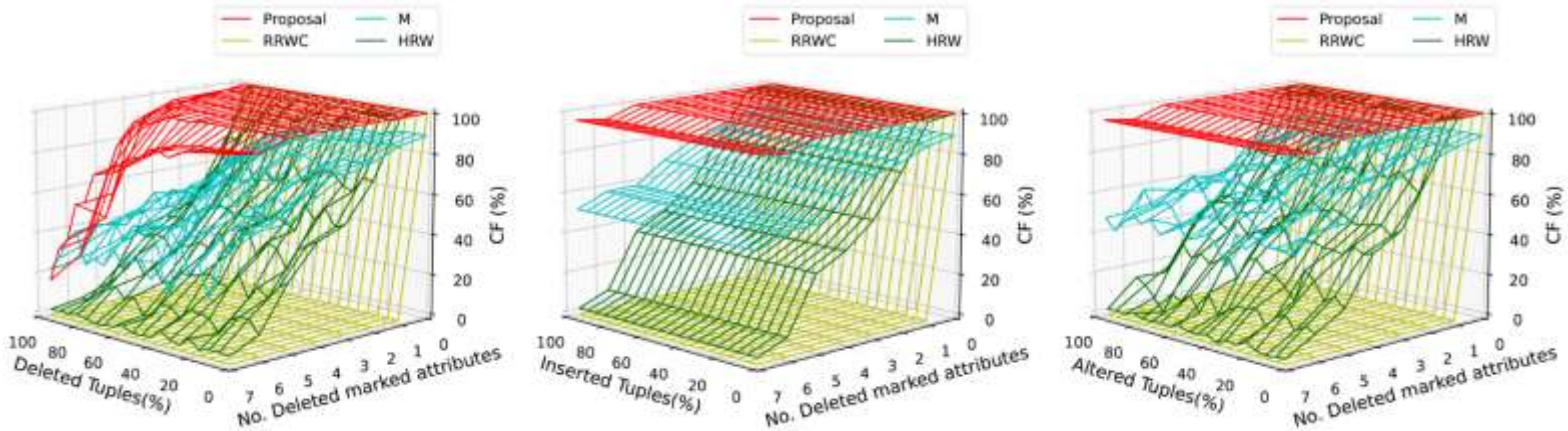
Algorithm	VPK	Unique VPK	ω_{max}	ω_{min}
Proposal	34761	3340	2331	689
M	3191	83	1326	0
HRW	2872	1181	161	33
RRWC	2906	2904	78	78

Normal single attack robustness verification

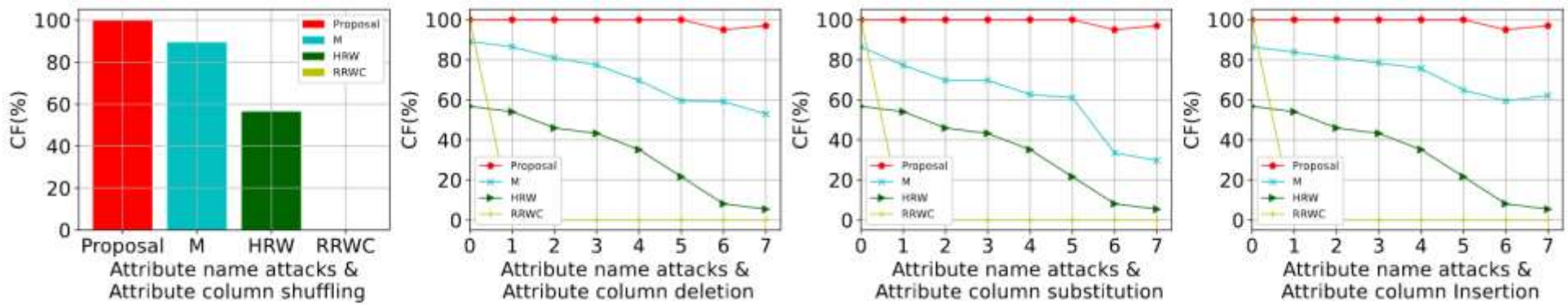


Experiments

Multifaceted attack robustness verification



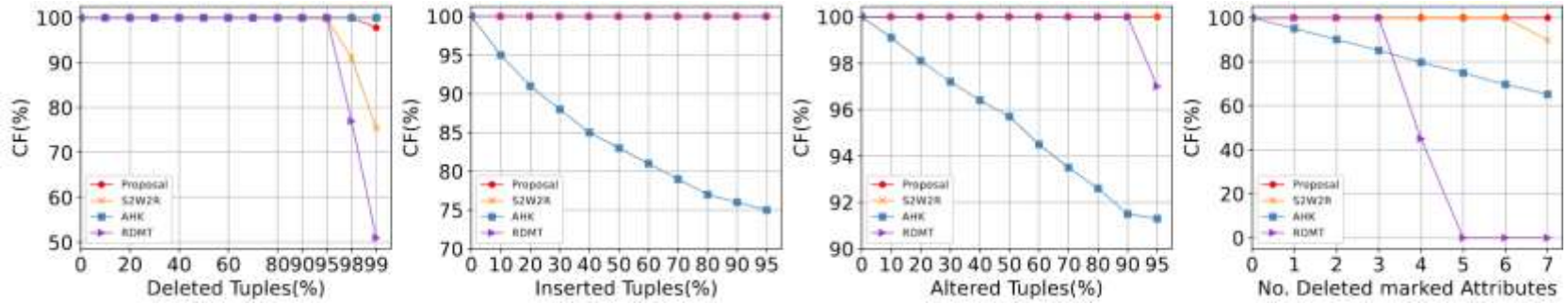
Mix-match attribute attack robustness verification



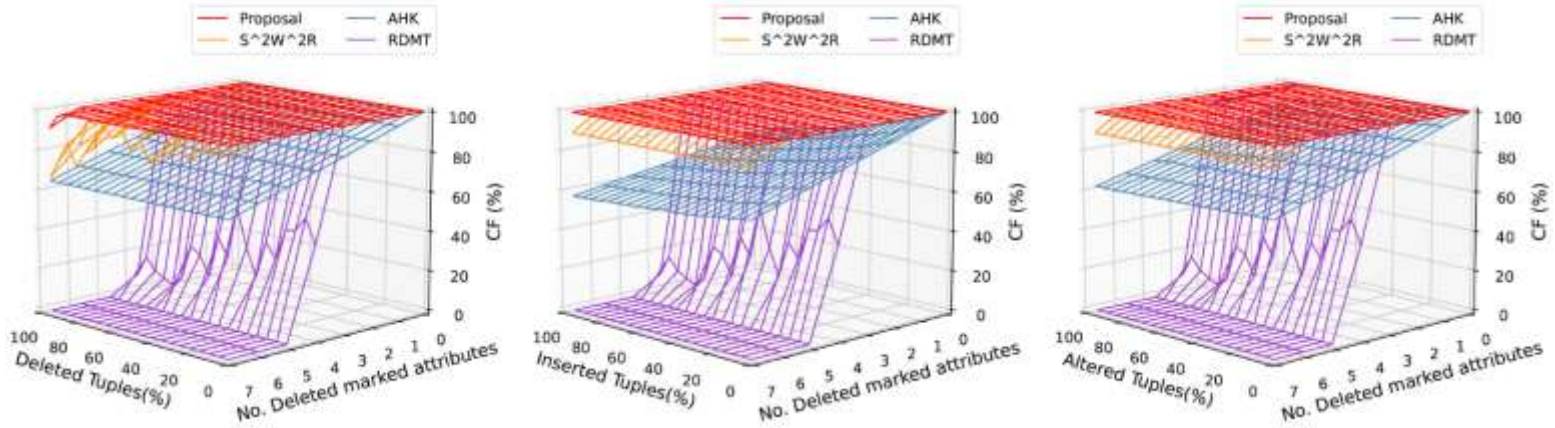
Experiments

Robustness Verification: PK watermarking algorithm

Normal single attack robustness verification



Multifaceted attack robustness verification



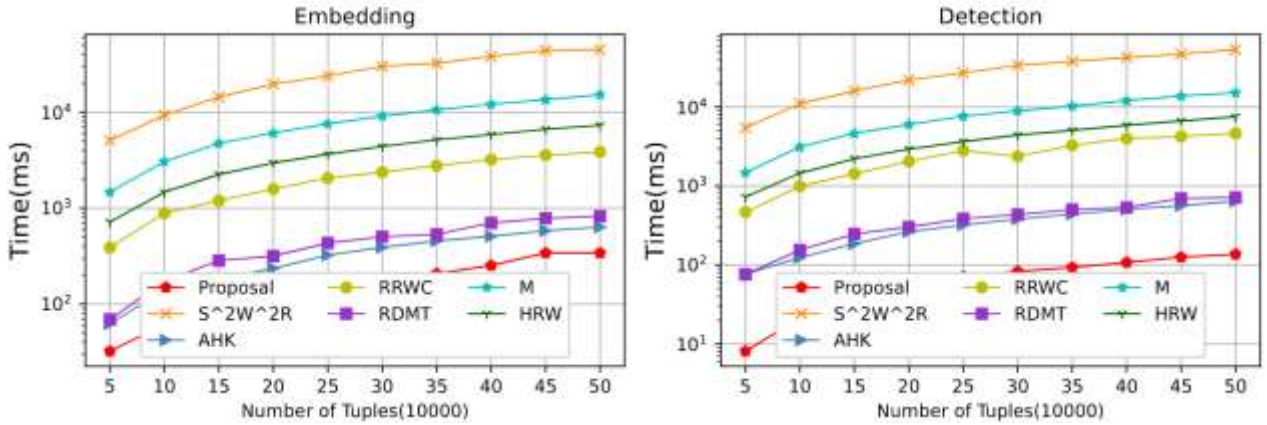
Experiments

Usability Verification

Data Distortions

Attribute	$\Delta\mu$	$\Delta\sigma$
Elevation	-1.89e-4	2.33e-4
Slope	-1.6e-4	7.98e-05
Horizontal_Distance_To_Hydrology	1.31e-4	-2.57e-4
Vertical_Distance_To_Hydrology	1.39e-4	-6.24e-6
Horizontal_Distance_To_Roadways	1.08e-4	-4.66e-7
Hillshade_9am	8.9e-5	8.56e-5
Hillshade_3pm	-9.8e-5	2.78e-5
Horizontal_Distance_To_Fire_Points	4.6e-5	1.81e-4

Overhead



Conclusion

- **Introduce Attribute column attacks and mix-match attribute attacks.**
- **Train an attribute classifier for resisting these attacks.**
- **Design a high-synchronization and robust virtual primary key scheme for watermarking relational databases.**
- **Performance experiments substantiate our scheme.**

Thanks

Any concern feel free to reach me out:
yangke@iie.ac.cn