Enhancing XML Zero-Watermarking Robustness With Usability Queries

Benedek Székács
b.szekacs-2@student.tudelft.nl

Supervisors

Dr. Zeki Erkin, z.erkin@tudelft.nl
Devris Isler, d.isler@imdea.org

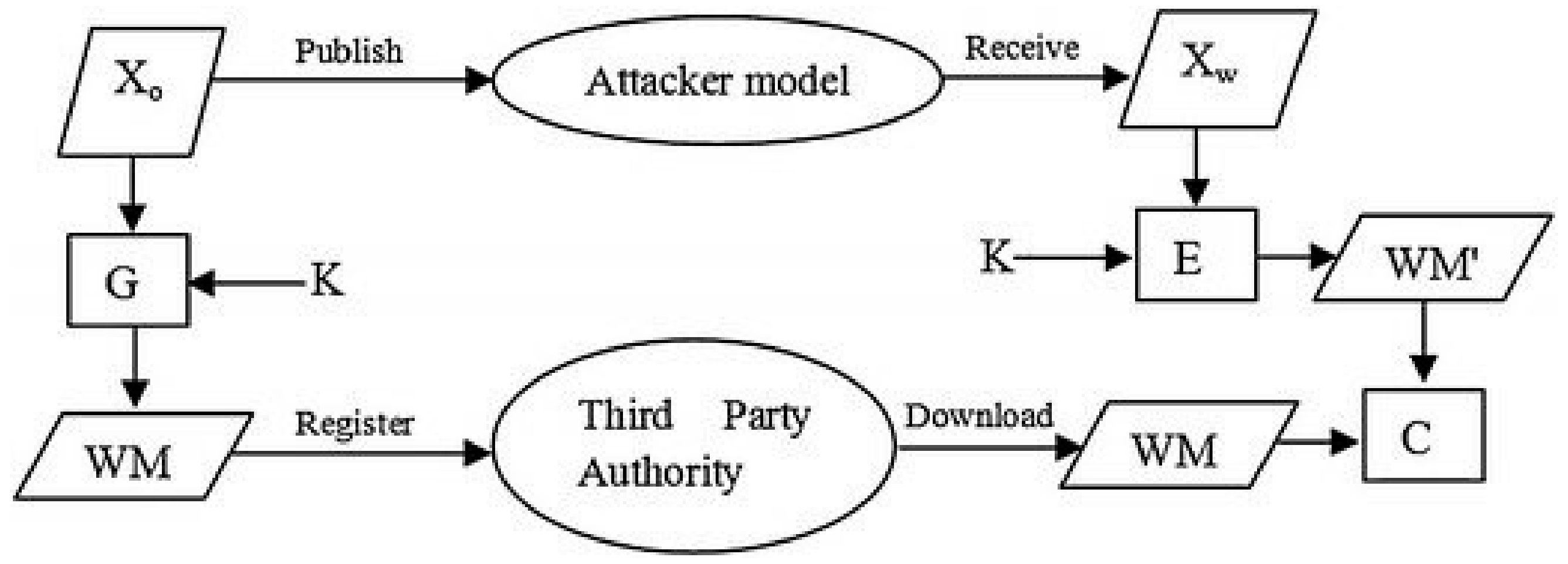
TU Delft

Introduction

- Each year the amount of **XML data** being exchanged over the internet greatly increases.
- Ensuring data intergrity and ownership is critical
- Relational watermarking techniques face challenges with XML's hierarchical nature.
- Wen et al. [1] proposes a zero-watermarking method, using functional dependencies in XML
- We improve robustness against zero-out and context specific attacks by integrating usability queries

Zero-Watermarking

- Distortion free - No embedding process
- Watermark is generated from the structural or semantical features of the data.
- Detection process is the same as the generation process, the two watermarks are compared to prove ownership

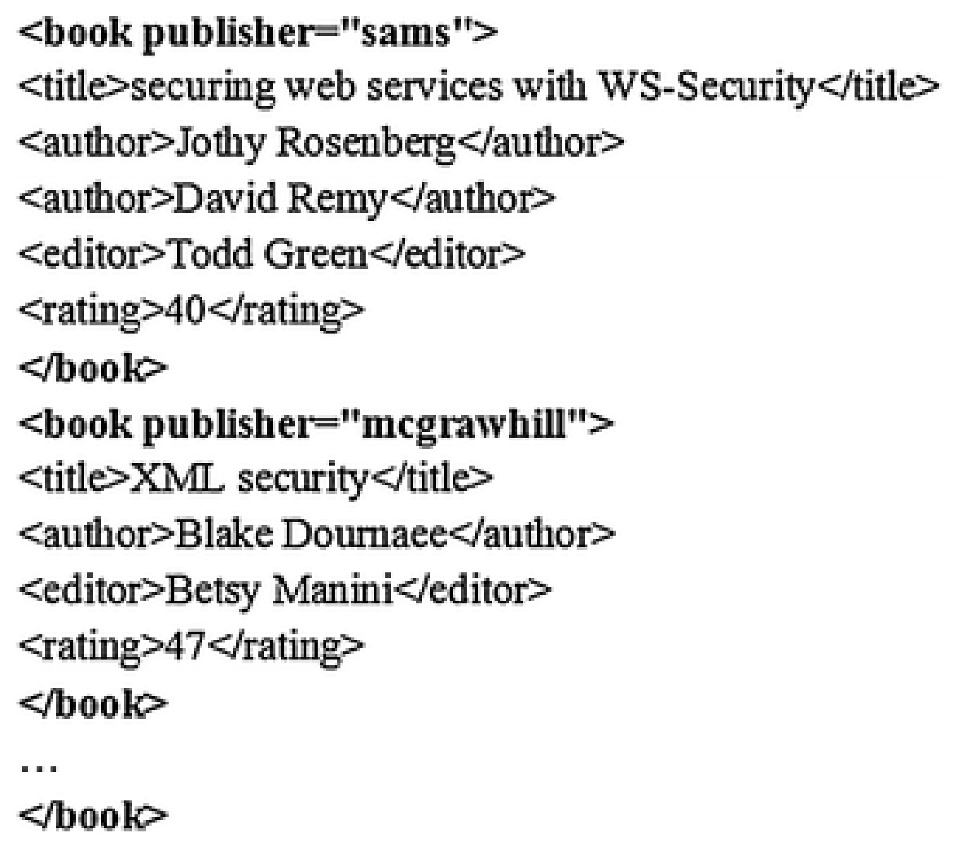


Functional Dependencies and Usability

- **Functional Dependency:** Relationships in XML data where certain values uniquely determine other values.
- *book/editor* → *book/publisher*

- Usability is defined by **query templates**
- *book[author]/title* is a template for querying books by author

- **Cover Range:** all paths covered by template
- From a set of usability query templates, the cover range includes all important attributes for data usability



```
<book publisher="sams">
<title>securing web services with WS-Security</title>
<author>Jothy Rosenberg</author>
<author>David Remy</author>
<editor>Todd Green</editor>
<rating>40</rating>
</book>
<book publisher="mcgrawhill">
<title>XML security</title>
<author>Blake Dournaee</author>
<editor>Betsy Manini</editor>
<rating>47</rating>
</book>
...
</book>
```

Procedure

- **DiscoverFD Algorithm:** Traverses the lattice of attribute sets to discover all intra-relation FDs and Keys.
- **Generation:** Extract functional dependencies using DiscoverFD, convert them to a binary string representation and encode them using the secret key to produce the watermark bits.
- **Detection:** Repeat the generation steps to get *WM'*. Compare with original watermark to get the similarity - detection rate.

- **Integrating Usability:**
  - Generate the cover range of user defined query templates.
  - Filter attributes in DiscoverFD to only include the ones given by the cover range.

- The DBLP computer science publication dataset is used, focusing only on '*inproceedings*' elements for simplicity.
- Implemented both algorithms in python.

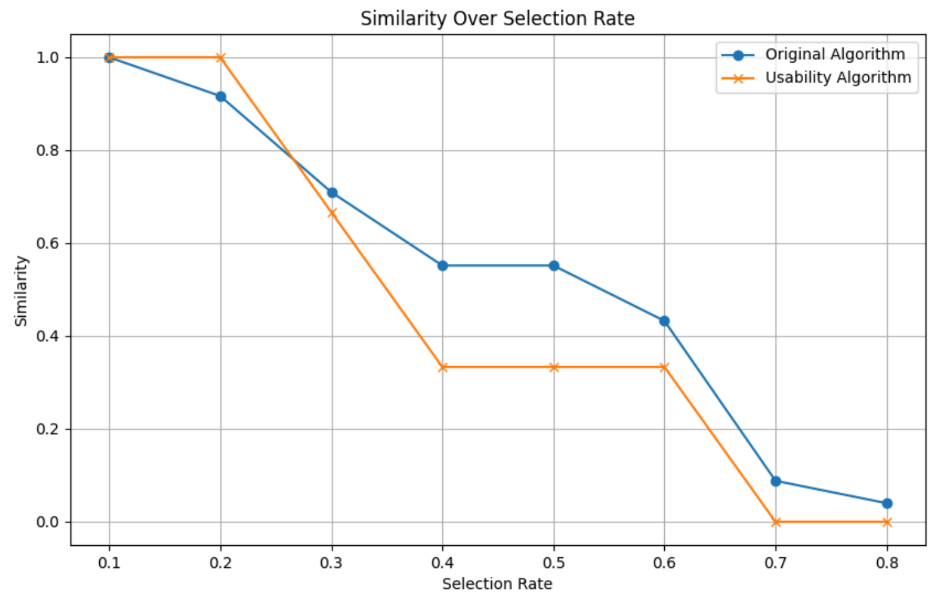
- We adopt the query templates from Zhou et al. [2] to describe the usability of DBLP:
  - *inproceedings[title]/author*
  - *inproceedings[author]*
  - *inproceedings[conference]/title*
- To increase watermark capacity, we added the following
  - *inproceedings[title]/year*
  - *inproceedings[booktitle]/title*

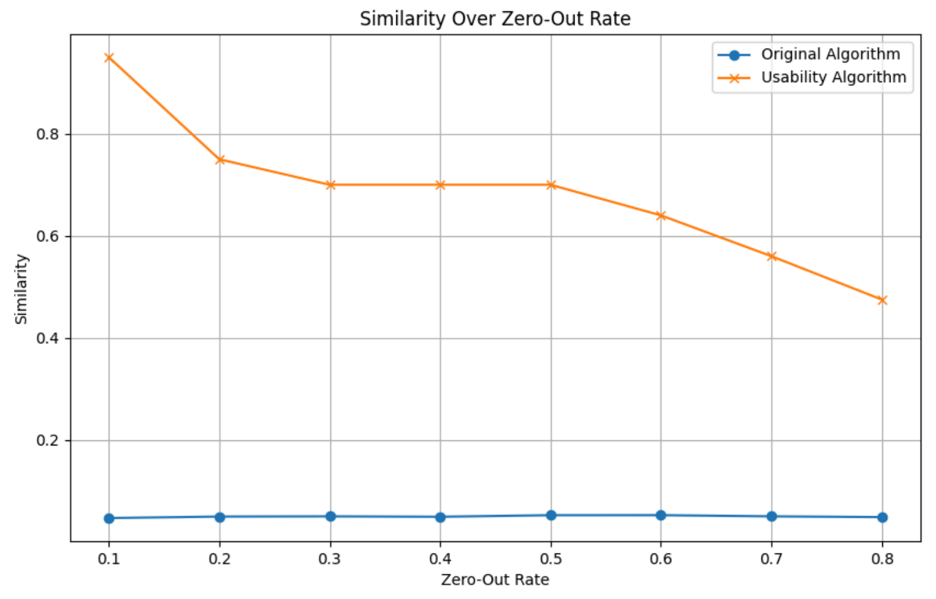
- We execute multiple attack types and measure the achieved similarity between the original and attacked watermarks.
- Simulate the first four attacks with random selection using varying attack ranges.
- All attacks are run 10 times and the average similarity is calculated.

Results - Standard Attacks

- **Selection Attack:** Randomly deletes a percentage of elements.
- The original algorithm achieves higher similarity with attack rates over 30%, due to it's higher watermark capacity.

- **Zero-Out Attack:** Changes attribute values of random nodes to zero.
- The original algorithm is extremely fragile against zero-out.
- By making at least one attribute uniform in value, a large number of extra FDs are introduced, destroying the watermark.

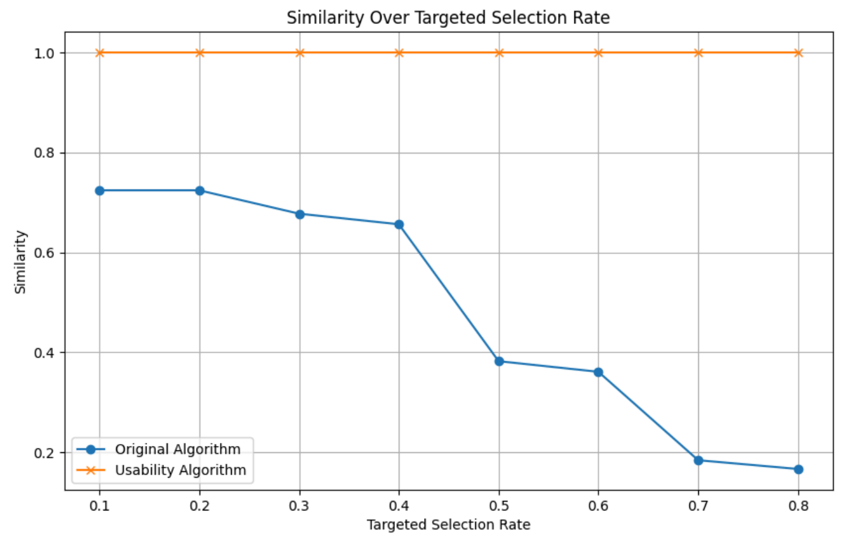

Similarity Over Selection Rate

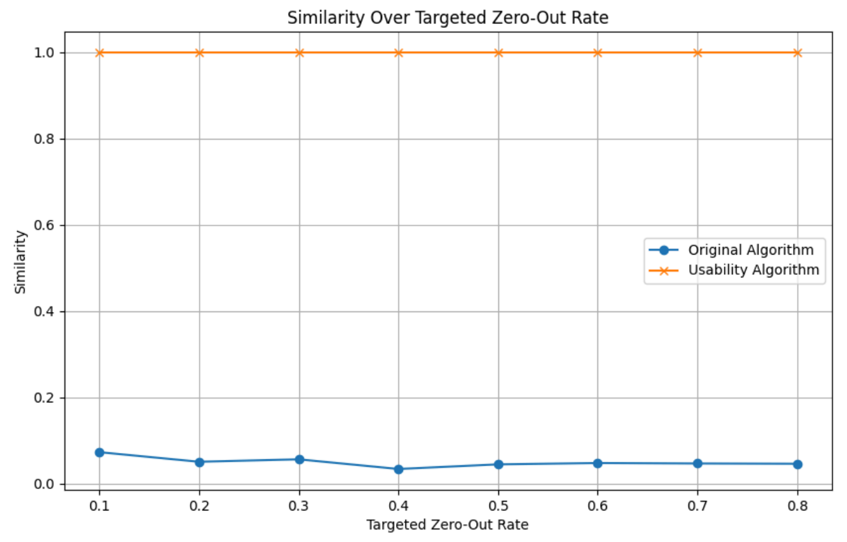

Similarity Over Zero-Out Rate

Results - Targeted Attacks

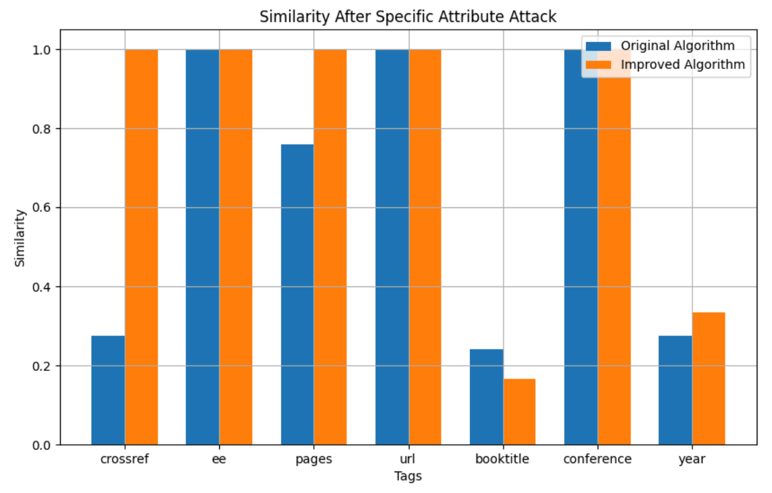
- **Targeted Selection Attack:** Deletes attributes of selected nodes based on their usability.
- Since the attack uses the same query templates to select unwanted attibutes, our method achieves maximal similarity for this experiment. The original algorithm performs worse than in standard selection, as a large part of the watermark is created from the targeted attributes.

- **Targeted Zero-Out Attack:** zeroes out values of attributes deemed unimportant for usability.
- Here we show that the original algorithm doesnt address any type of zero-out attacks, and our method still achieves maximum similarity due to the attacker usability definition.


Similarity Over Targeted Selection Rate


Similarity Over Targeted Zero-Out Rate

- **Single Attribute Selection:** deletes the least significant attribute from the schema in terms of usability.
- The results show what attributes contribute to the functional dependencies in the data.
- Our algorithm achieves a similarity of 1 for attack against attributes outside of the cover range.


Similarity After Specific Attribute Attack

Future work

- **Diverse Dataset Evaluation:** Apply the proposed zero-watermarking method to various XML datasets with different schemas and sizes to evaluate its generalizability and performance across different contexts.
- **Advanced Attack Simulations:** Explore and simulate more sophisticated attack types, including those that specifically target non-essential attributes, to identify potential vulnerabilities and develop robust defenses.
- **Hybrid Watermarking Method:** Develop a hybrid watermarking approach that balances usability and watermark capacity, combining the strengths of usability-driven and traditional watermarking techniques to achieve better overall performance.

References

[1] Zhong Wen, Xiangliang Wang, and Yongjian Li. Zero-watermarking for xml data based on functional dependencies. Journal of Real-Time Image Processing, 13(2):313–324, 2016.
[2] Xuan Zhou, HweeHwa Pang, and Kian-Lee Tan. Query-based watermarking for xml data. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pages 437–448, 2007.