TAP: Transparent and Privacy-Preserving Data Services

Daniël Reijsbergen\textsuperscript{1}  Aung Maw\textsuperscript{2}  Zheng Yang\textsuperscript{3}
Tien Tuan Anh Dinh\textsuperscript{4}  Jianying Zhou\textsuperscript{2}

\textsuperscript{1}Nanyang Technological University, Singapore
\textsuperscript{2}Singapore University of Technology and Design, Singapore
\textsuperscript{3}Southwest University, China
\textsuperscript{4}Deakin University, Australia

USENIX Security 2023, Anaheim, CA, USA
11 August 2023
Data Services

• Companies gather data from users, perform computations

• Example: Smart Grid
  • What is the total energy use in my area?
  • What are the average and standard deviation?
  • What is the maximum among residential users?
  • What is the 95% quantile?

• Other Examples:
  • Congestion pricing
  • Digital advertising
Transparency

- *Challenge:* companies may have a financial incentive to *cheat*

- We want to guarantee the following:
  - *Data Integrity:* data is not tampered with
  - *Transparency:* computations on data are performed correctly
  - *Data Privacy:* users cannot view data values of other users

- Also: rich set of operations (sums, quantiles, ...), *efficiency*
TAP

• Naïve solutions:
  • All data on company server: *privacy*, no *transparency*
  • All data public: *transparency*, no *privacy*

• Other existing approaches are insufficient:
  • *Limited query support* (e.g., transparency logs, proofs-of-liabilities), or
  • *Single-user* (e.g., authenticated databases)

• **TAP**: a verifiable log with rich query support
TAP: System Model

**Users:**
- Monitor their data values
- Perform queries

**Server:**
- Builds data structure
- Generates responses, proofs

**Auditors:**
- Check data structure
TAP: Data Structure

- Two-layer structure: **prefix tree**, with a **sum tree** in each leaf
- Sum tree leaf for each **data value**: max. 1 value per user per time slot

![Diagram of TAP data structure](image)
TAP: Prefix Tree

- One prefix tree leaf for each combination of attributes
- Top tree is chronological ⇒ append to the right, easy to audit
TAP: Sum Trees

- Nodes store *hom. commitments* of values and higher stat. moments
- Leaves are *sorted*: audited using *zero-knowledge (zk) proofs*
TAP: Performance

- Server can prove query correctness \textit{efficiently}
  - Sum/average using sum tree roots
  - Min/max/quantiles using \textit{zk-proofs} and \textit{sorted} leaf structure

- Practical performance on $1$/hour Amazon machine:
  - Smart grid with 1.8 million users, 100 sum trees / time slot:
    - less than 5 minutes to update tree
  - Max. audit volume: 360 000 values per hour
Conclusion

• **TAP** uses *two-layer* tree structure and *zero-knowledge proofs*
• Guarantees *integrity, transparency, and privacy*
• **Verifiable log** with rich query support: sum, variance, quantiles, ...

• Future work:
  • More query types (e.g., correlation)
  • Improve efficiency when most data values are zero
  • Implement extension to *differential privacy*
Thank You!

Please contact us via email:

daniel.reijsbergen@ntu.edu.sg
youngzheng@swu.edu.cn
anh.dinh@deakin.edu.au
jianying_zhou@sutd.edu.sg