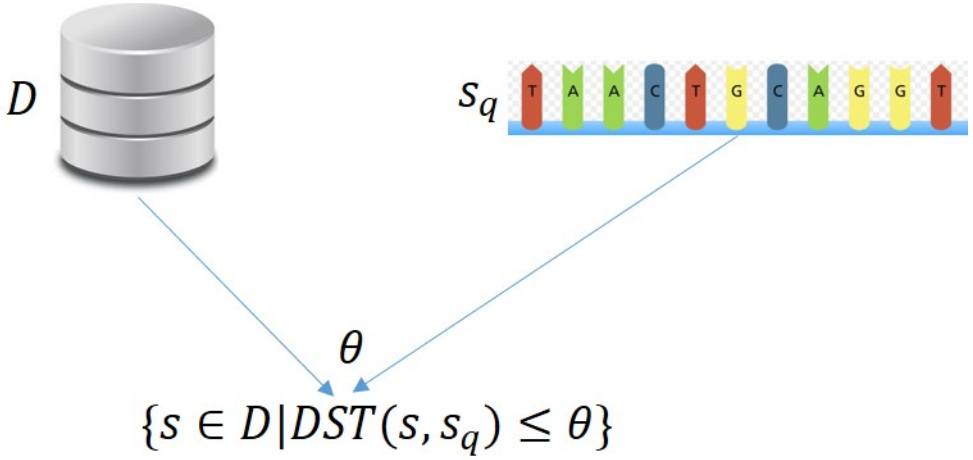
ARM: Authenticated Approximate Record Matching for Outsourced Databases

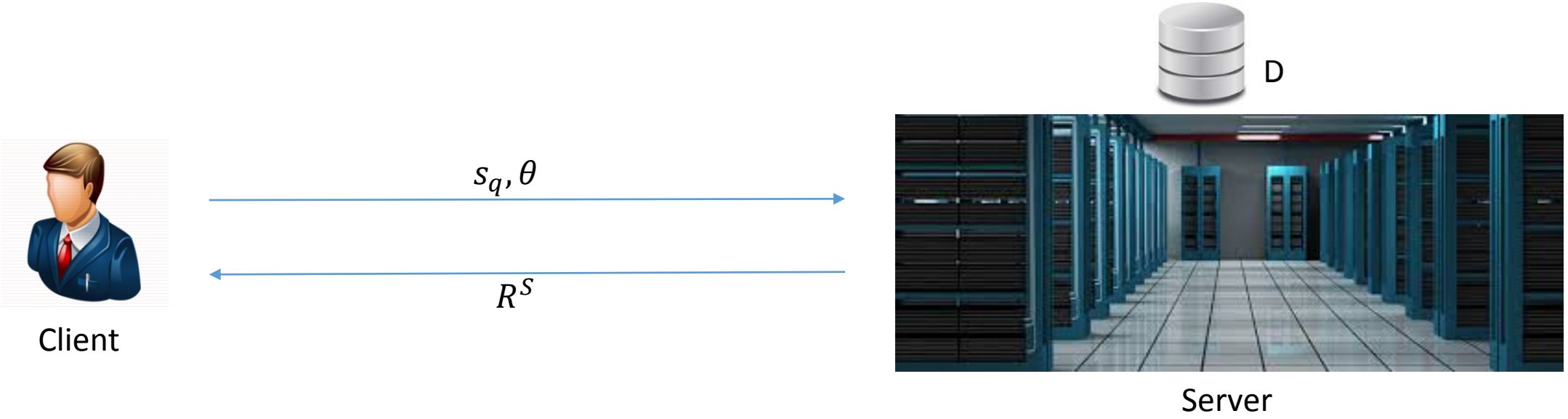
Boxiang Dong, Wendy Wang
Stevens Institute of Technology
Hoboken, NJ

Approximate Record Matching



- D: dataset
- s_q: target record
- $DST()$: distance function
- θ: distance threshold

Outsourcing Framework



- D: dataset
- s_q: target record
- θ: distance threshold
- R^S: similar records returned by the server

Attack Model

- The untrusted server may not execute the computation honestly.
  - Generate cheaper results to save computational cost.
  - Intentionally return incorrect result for competition reason.

- The returned similar records R^S may not be sound or complete.
  - Soundness: all records in R^S are similar to s_q.
  - Completeness: all similar records are included in R^S.

Result Authentication

Untrusted server may

- Tampered values: $s \in R^S, s! \in D$
- Result soundness violation: include dissimilar record $s \in R^S, DST(s, s_q) > \theta$
- Result completeness violation: remove similar record: $s! \in R^S, DST(s, s_q) \leq \theta$

- Naive solution
  - The client obtains R by record matching locally, and compares it with R^S.
  - Computationally expensive.

Verification Goal

An authentication framework

- catches
  - Tampered values
  - Result soundness violation
  - Result completeness violation
- is scalable and efficient
- supports data updates

Related Work

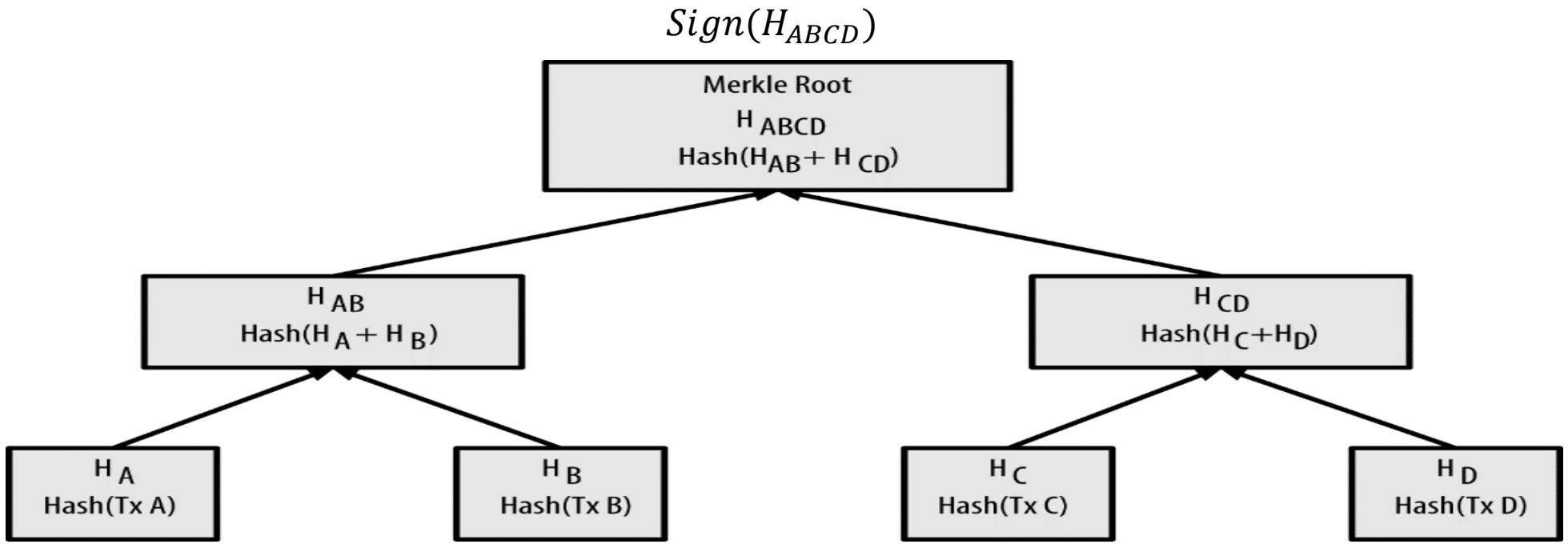
- Privacy-preserving record matching [1, 2, 3]
  - Prevents the server from possessing the original data.
- Authentication of SQL query [4, 5]
  - Cannot be applied to approximate matching.
- Authentication of keyword search [6, 7]
  - Only supports exact match.
  - Does not support data update.

Our Contributions

- We design an authentication tree structure named *MB-tree*.
  - It is not limited to main memory usage.
  - It supports incremental data updates.

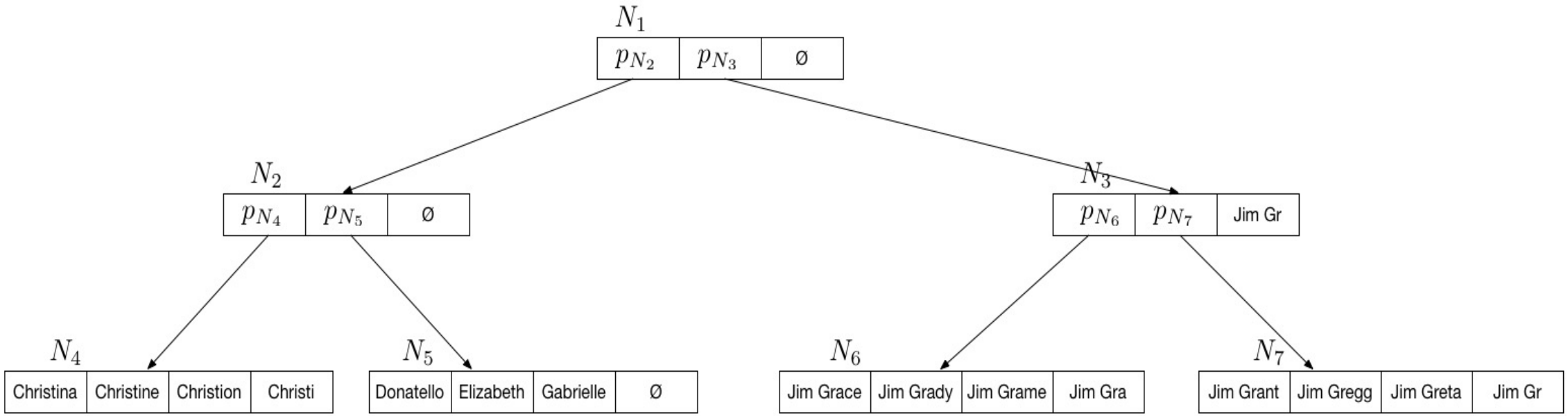
- We propose a lightweight authentication method.
  - The server constructs the verification object (VO) of the returned results with the MB-tree.
  - The client verifies the soundness and completeness of the returned results by using VO.

Preliminaries: Merkle Tree [9]



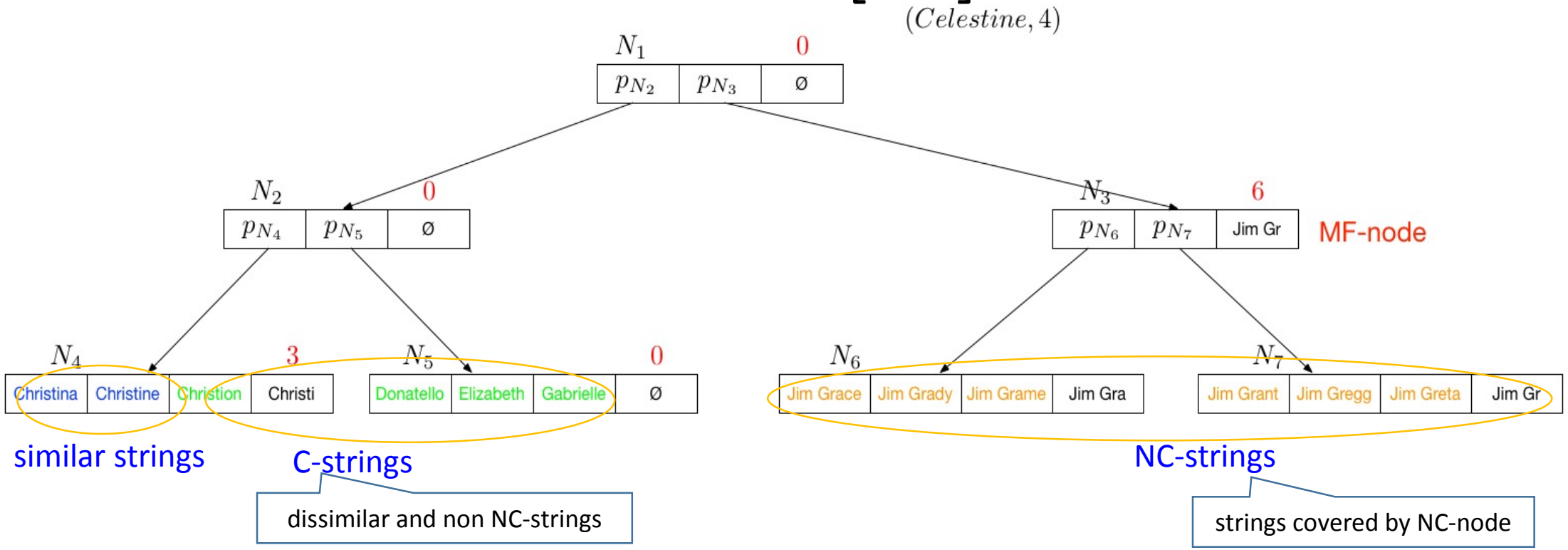
- An authentication data structure based on hash function.
- Hash is much more efficient than edit distance computation.

Preliminaries: B^{ed}-Tree [10]



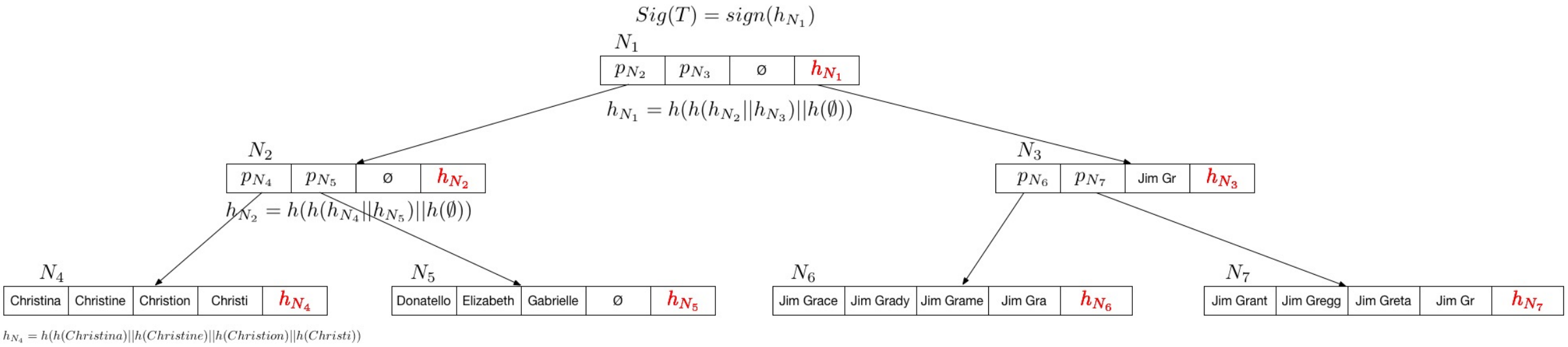
- Sort the records in dictionary order.
- Store the longest common prefix (LCP) of the enclosed strings.

Preliminaries: B^{ed}-Tree [10]



- Avoid the edit distance calculation for the NC-strings.
- Performs well with
  - Memory constraints
  - Data updates.

MB-tree (Ours)



- The client constructs a MB-tree (Merkle B^{ed}-tree).
- Only the signature is kept locally.

Authentication Framework

$T = MBTree(D)$
$Sig(T) = sign(root)$

T →

D

Client

s_q, θ →

$VO = traverse(R^S, T, s_q, \theta)$

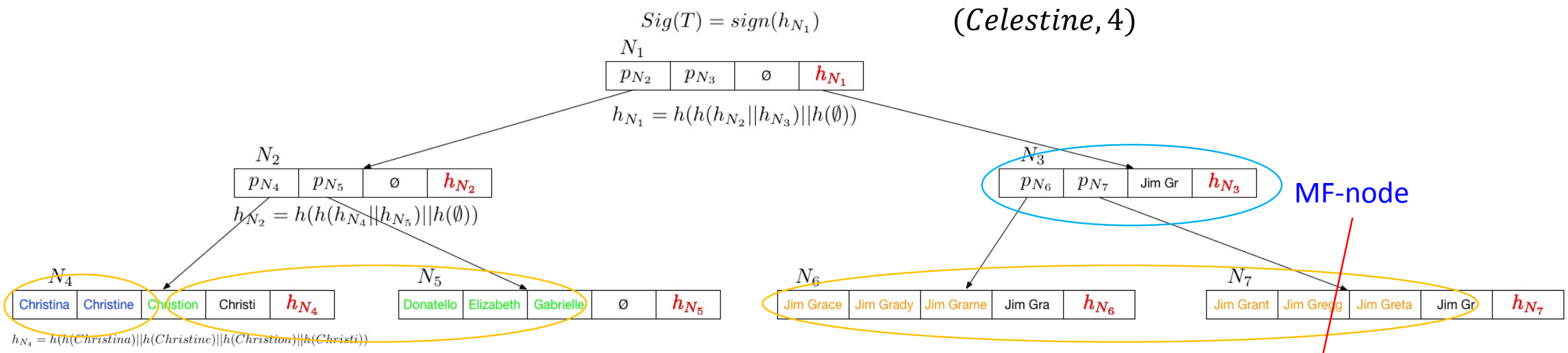
← R^S, VO

Server

$Verify(R^S, VO)$

- D: dataset
- T: MB-tree
- VO: verification object
- s_q: target record
- θ: distance threshold
- R^S: similar records returned by the server

VO Construction for A Single Target Record



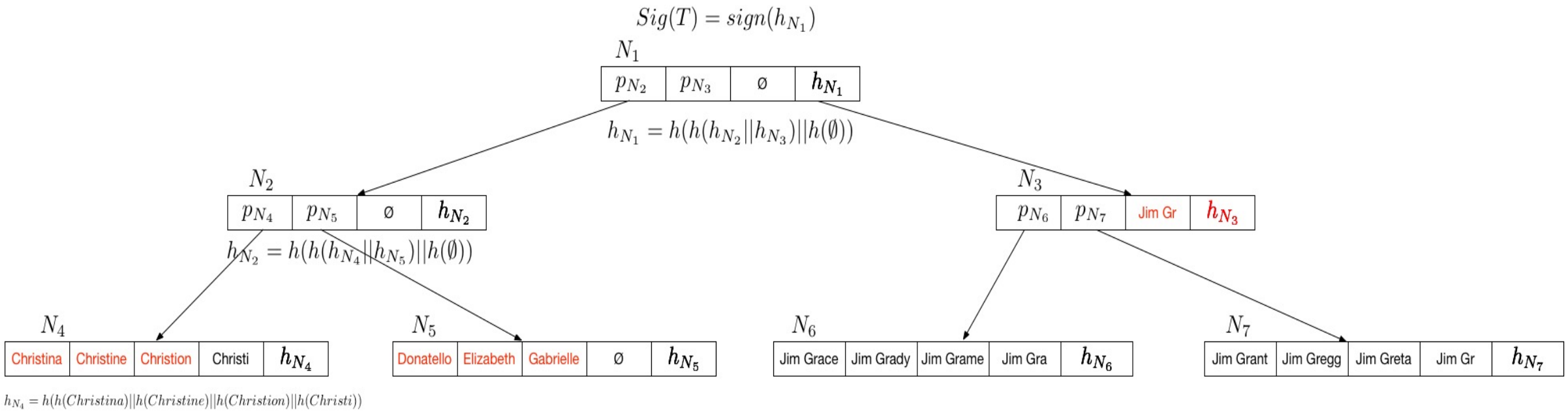
$$R^S = \{Christina, Christine\}$$

$$VO = \{(((Christina, Christine, Christion), (Donatello, Elizabeth, Gabirelle)), (Jim\ Gr, h_{N_3}))\}$$

- Key idea: use the MF-node to substitute the large amount of NC-strings.

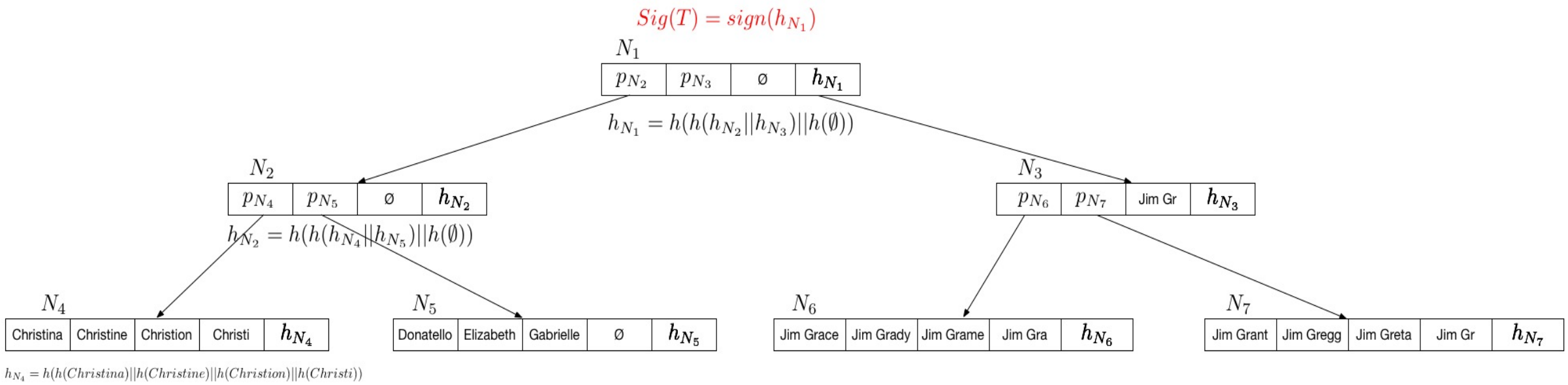
Verification

- Check tampered values: re-compute root signature of MB-tree



Verification

- Check tampered values: re-compute root signature of MB-tree



- Check if Sig(T) matches the local copy of signature.

Verification

Similarity verification

- Result soundness check: $\forall s \in R^S$, check if $DST(s, s_q) \leq \theta$.
- Result completeness check
  - $\forall$C-string s, check if $DST(s, s_q) > \theta$.
  - $\forall$MF-node N, check if $MIN\_DST(N, s_q) > \theta$.

Verification

$R^S = \{Christina, Christine\}$

$VO = \{(((Christina, Christine, Christion), (Donatello, Elizabeth, Gabirelle)), (Jim\ Gr, h_{N_3}))\}$

Edit Distance Computation

Soundness check: $\forall s \in R^S$, check if $DST(s, s_q) \leq \theta$.

$DST(Christina, Celestine) = 4$

$DST(Christine, Celestine) = 3$

Completeness check

$\forall$C-string s, check if $DST(s, s_q) > \theta$.

$DST(Christion, Celestine) = 5>4$

$DST(Donatello, Celestine) = 9>4$

$DST(Elizabeth, Celestine) = 9>4$

$DST(Gabirelle, Celestine) = 8>4$

$\forall$MF-node N, check if $MIN\_DST(N, s_q) > \theta$.

$MIN\_DST(Jim\ Gr, Celestine) = 6 > 4$

7 DST calculations

Naïve method: 12 DST calculations

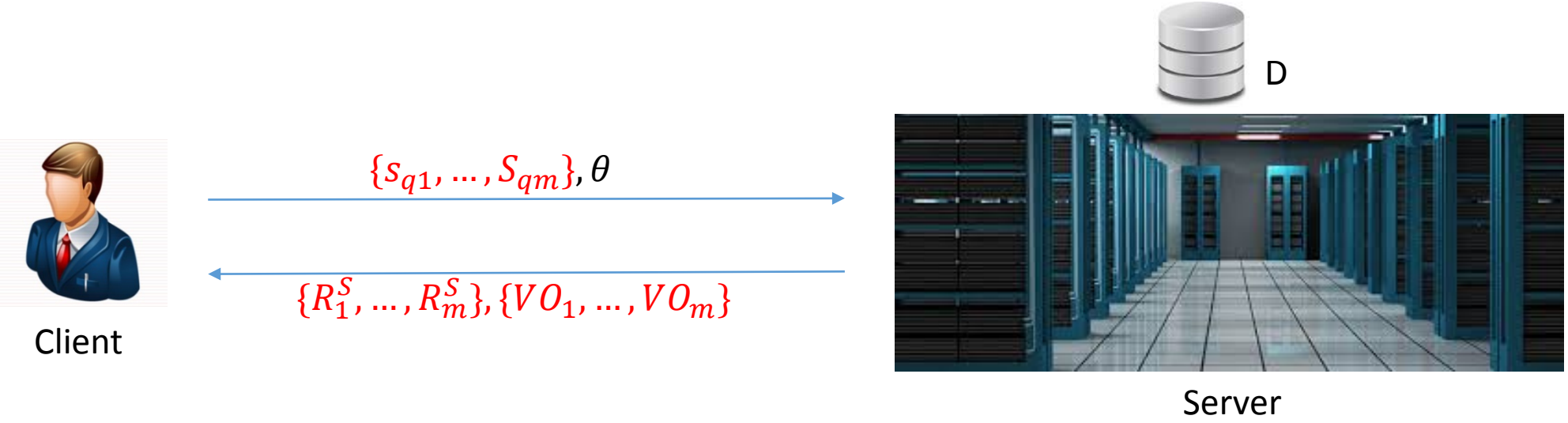
Complexity

- Pre-processing
  - Time: $O(n)$
  - Space: $O(1)$
- VO construction
  - Time: $O(n)$
  - Space: $O\big((n_C+n_R)\sigma_s + n_{MF}\sigma_M)$
- Verification
  - Time: $O((n_R + n_{MF} + n_C)C_{Ed})$

n: # of strings in D
n_C: # of C-strings
n_R: # of similar strings
n_{MF}: # of MF-nodes

σ_s: avg. length of string
σ_M: size of MB-tree node
C_{Ed}: string edit distance complexity

VO Construction for Multiple Target Records



- So far, we only consider the authentication for a single target record.
- We have optimizations for multiple target records
  - optimization based on triangle inequality
  - optimization based on overlapped dissimilar strings
- Details are in the paper.

Security

- Tampered values: Sig(T) does match the local copy.
- Result soundness violation: $\exists s \in R^S$ s.t. $DST(s, s_q) > \theta$
- Result completeness violation
  - $\exists$C-string s s.t. $DST(s, s_q) \leq \theta$.
  - $\exists$MF-node N s.t. $MIN\_DST(N, s_q) \leq \theta$.

Experiments

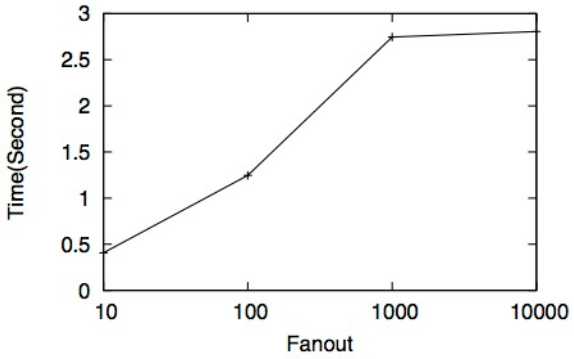
- Datasets

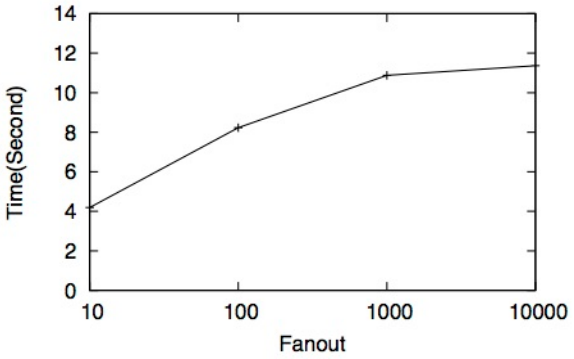
Dataset	# of records	Avg. record length
Actors (IMDB)	260,000	14.627
Authors (DBLP)	1,000,000	14.732

- Setup
  - Implementation: C++
  - Testbed: 2.4 GHz CPU, 48 GB RAM, Linux 3.2

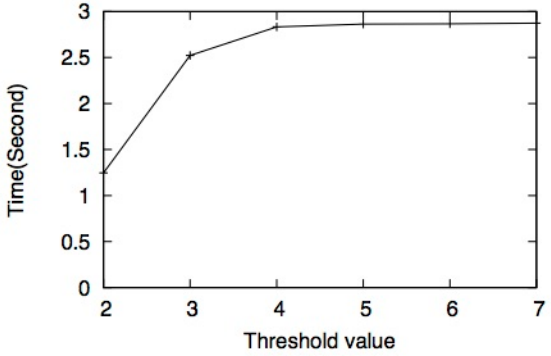
- Baseline: record matching locally.

VO Construction

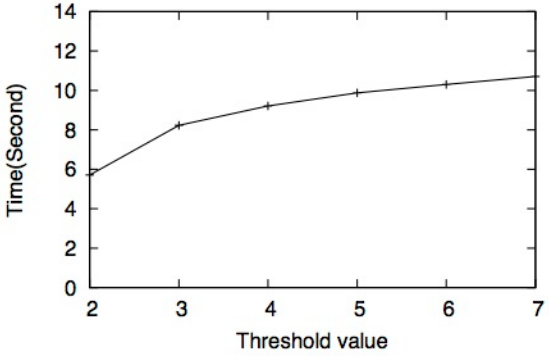




Small fanout leads to fast VO construction, as MIN_DST provides a tighter bound.



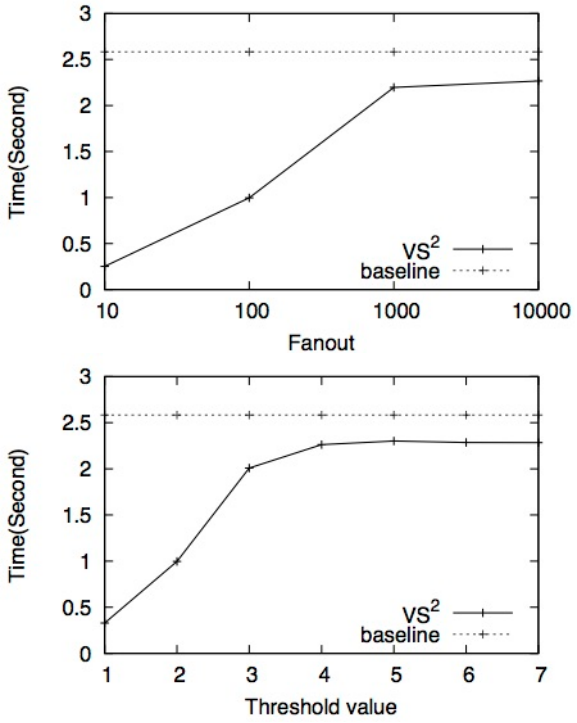
Actors dataset



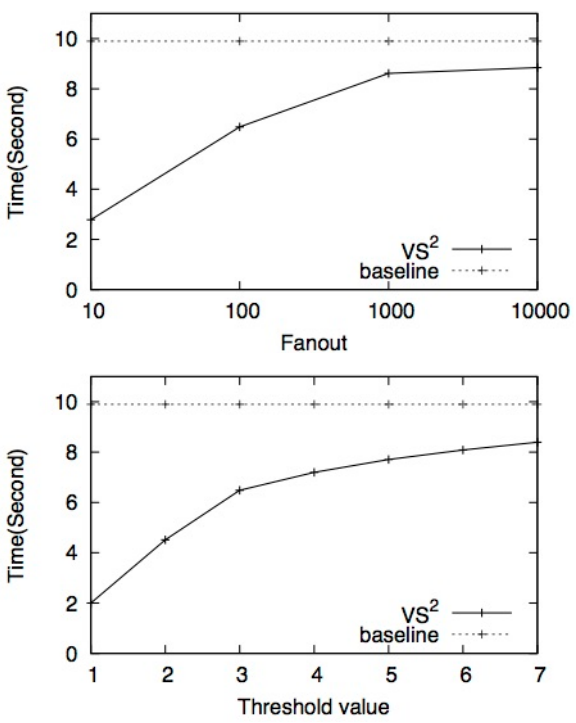
Authors dataset

Small threshold leads to fast VO construction, as there are more MF-nodes.

Verification



Actors dataset



Authors dataset

The verification is far more efficient than the baseline.

Conclusion

- We design an efficient authentication data structure – MB-tree based on B^{ed}-tree and Merkle tree.
- We propose efficient authentication framework for outsourced approximate record matching.
- Experimental results show the efficiency of our method.

Reference

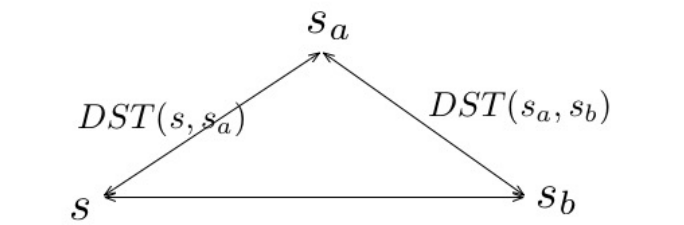
- [1] Hacigümüş, Hakan, et al. "Executing SQL over encrypted data in the database-service-provider model." Proceedings of the 2002 ACM SIGMOD international conference on Management of data. ACM, 2002.
- [2] Li, Feifei, et al. "Dynamic authenticated index structures for outsourced databases." Proceedings of the 2006 ACM SIGMOD international conference on Management of data. ACM, 2006.
- [3] Pang, HweeHwa, Jilian Zhang, and Kyriakos Mouratidis. "Scalable verification for outsourced dynamic databases." Proceedings of the VLDB Endowment 2.1 (2009): 802-813.
- [4] Goodrich, Michael T., et al. "Efficient verification of web-content searching through authenticated web crawlers." Proceedings of the VLDB Endowment 5.10 (2012): 920-931.
- [5] Pang, HweeHwa, and Kyriakos Mouratidis. "Authenticating the query results of text search engines." Proceedings of the VLDB Endowment 1.1 (2008): 126-137.
- [6] Durham, Elizabeth A., et al. "Composite bloom filters for secure record linkage." IEEE transactions on knowledge and data engineering 26.12 (2014): 2956-2968.
- [7] Schnell, Rainer, Tobias Bachteler, and Jörg Reiher. "Privacy-preserving record linkage using Bloom filters." BMC medical informatics and decision making 9.1 (2009): 41.
- [8] Zhang, Zhenjie, et al. "Bed-tree: an all-purpose index structure for string similarity search based on edit distance." Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.
- [9] Merkle, Ralph C. "A digital signature based on a conventional encryption function." *Conference on the Theory and Application of Cryptographic Techniques*. Springer Berlin Heidelberg, 1987.
- [10] Zhang, Zhenjie, et al. "Bed-tree: an all-purpose index structure for string similarity search based on edit distance." *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010.

Please contact:

bdong@stevens.edu Hui.Wang@stevens.edu

Optimizations for Multiple Target Records

- Optimization based on triangle inequality



$$DST(s, s_b) \leq DST(s, s_a) + DST(s_a, s_b) \leq \theta$$

- Optimization based on overlapped dissimilar strings