EARRING: Efficient Authentication of Outsourced Record Matching

IRI'17

Boxiang Dong [1] Wendy Wang [2]

[1]Montclair State University
Montclair, NJ

[2]Stevens Institute of Technology
Hoboken, NJ

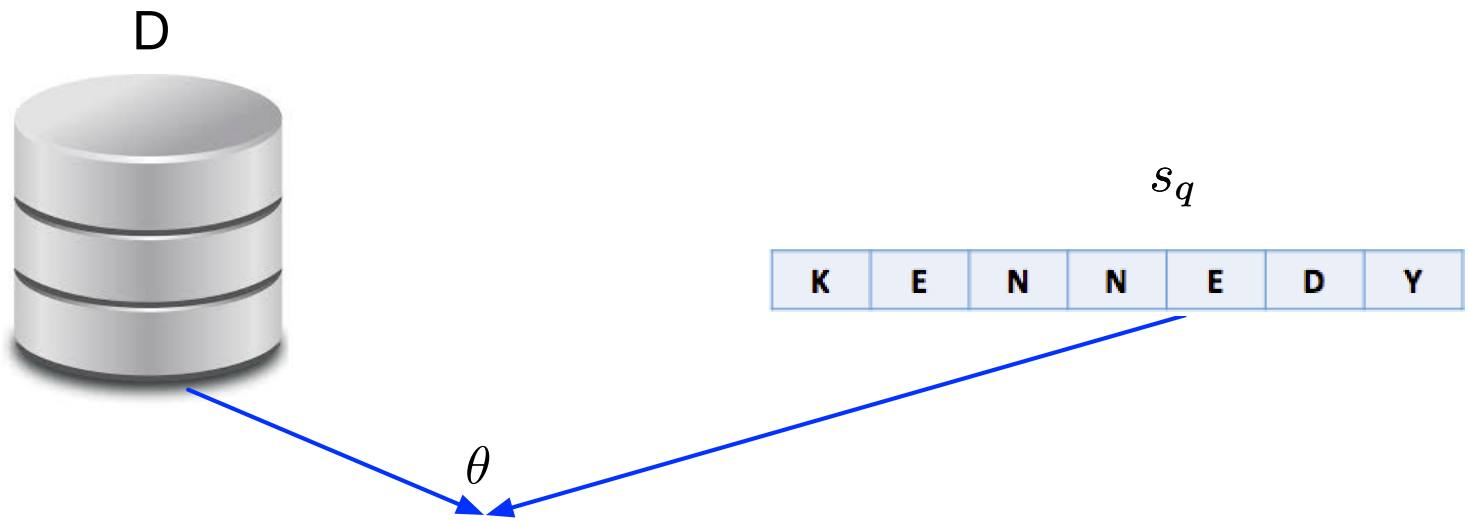
August 5, 2017

Record Matching

Record matching Find all records that are similar to a target.

Applications

- information integration
- data cleaning



$$\{s | s \in D, DST(s, s_q) \leq \theta\}$$

θ: similarity threshold
DST: edit distance

Record Matching

Record matching Find all records that are similar to a target.

Applications

- information integration
- data cleaning
- information retrieval

RID	Name	Street	City	Age
r_1	John	Leonard	NY	45
r_2	Kevin	Wicks	LA	31
r_3	Mike	Main	Phil	22

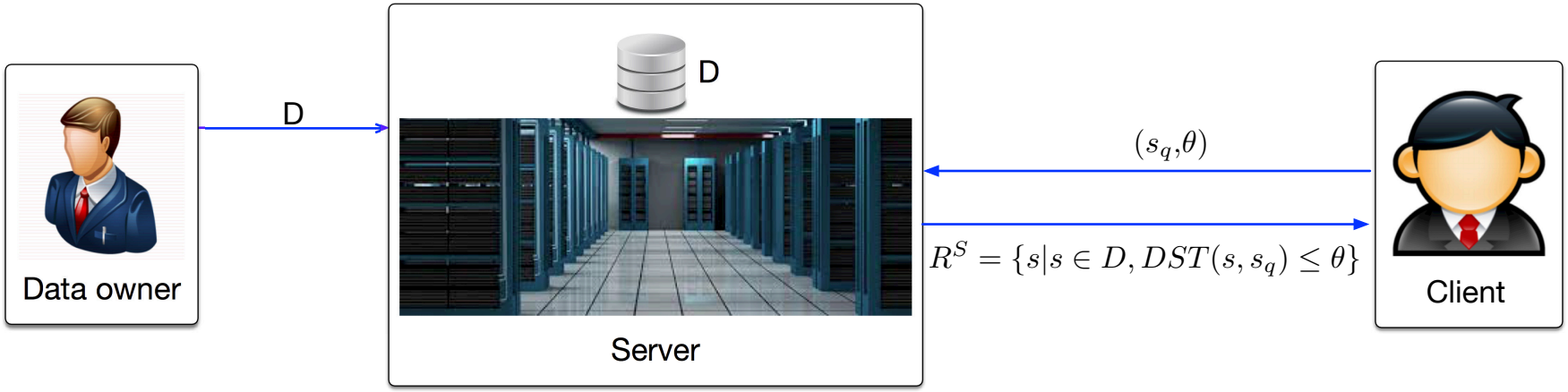
s_q = (John, Lenard, NY, 45)

$\theta = 2$

$\{r_1\}$

Outsourced Record Matching

- The third-party service provider (server) is responsible for processing the record matching requests.
- Outsourcing provides a cost-effective solution for the data owner.



Integrity Concern

- As the server is untrusted, it may return incorrect matching result.
- It is necessary to verify the *soundness* and the *completeness* of the result.

  Soundness $\forall s \in R^S$, $s \in D$ and $DST(s, s_q) \leq \theta$.
  Completeness $\forall s \in D$ s.t. $DST(s, s_q) \leq \theta$, $s \in R^S$.

Authentication Objective

We aim at an authentication framework that satisfies the following objectives.

- Authentication Objective
  - catches
    - soundness violation
      - $\exists s \in R^S$, but $s \notin D$
      - $\exists s \in R^S$, but $DST(s, s_q) > \theta$
    - completeness violation
      - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$
  - supports efficient verification
  - scales well with big data

Outline

Related Work

Privacy-preserving record matching

- Data encoding [DLW14]
- Secure multiparty computation protocol [KV15]

Authentication of outsourced SQL queries

- Hardware-based solution [BS13]
- Aggregation queries [NT05]
- Selection-projection queries [MNT06]

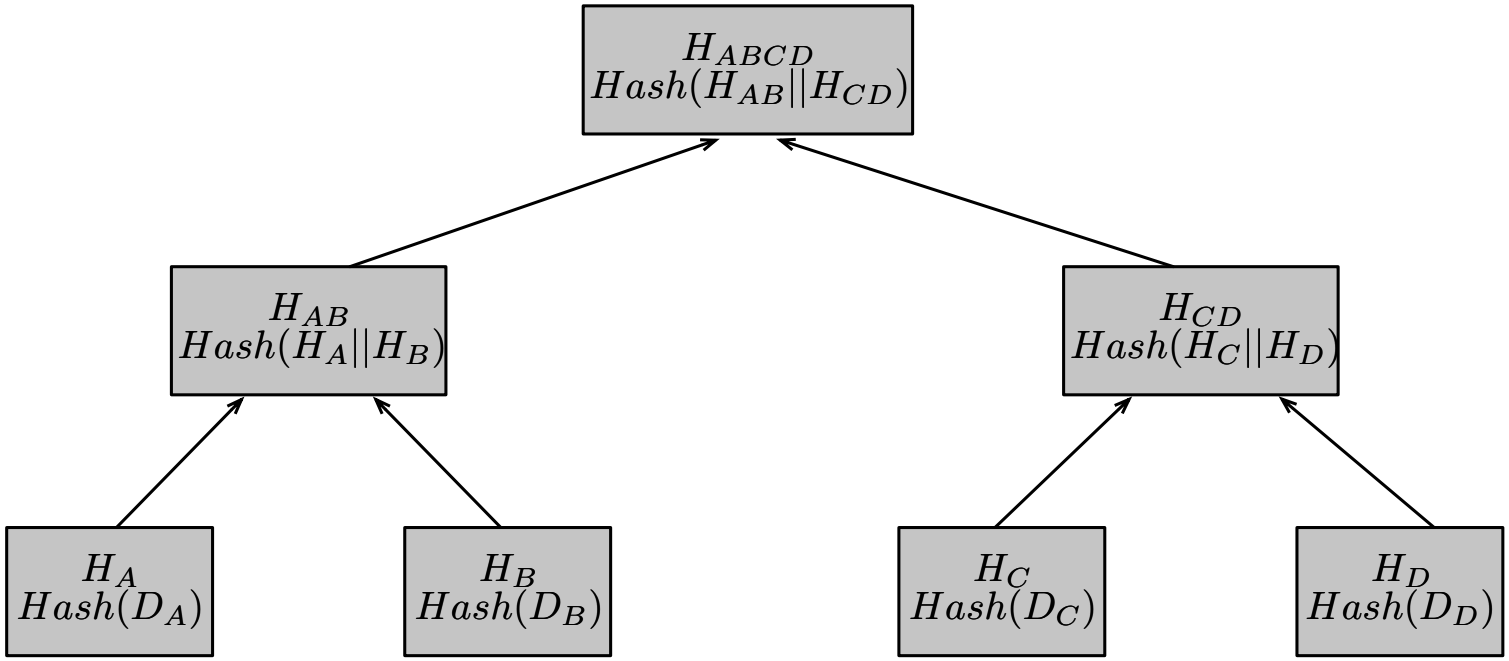
Authentication of nearest neighbor search

- Verification object (VO) [YPPK08, YLY11]

Outline

Merkle Tree

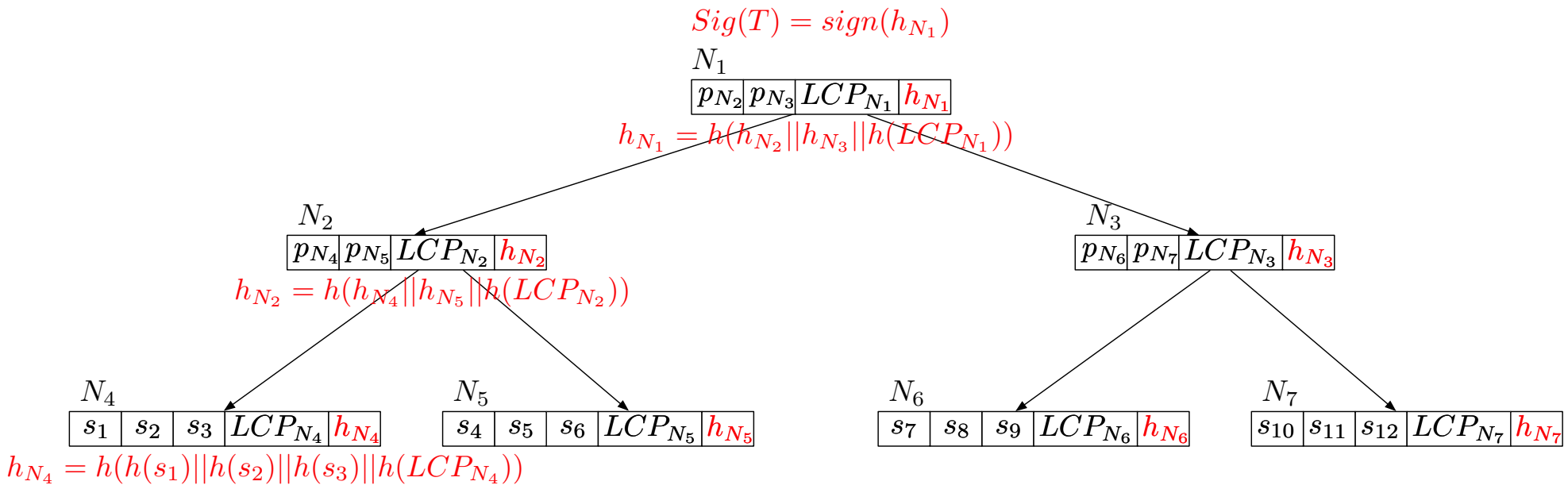
Merkle tree is a generalization of hash lists and hash chains.



- It allows efficient and secure verification of the contents of large data structures.
- Hash is computationally more efficient than edit distance calculation.

ARM [DW16]

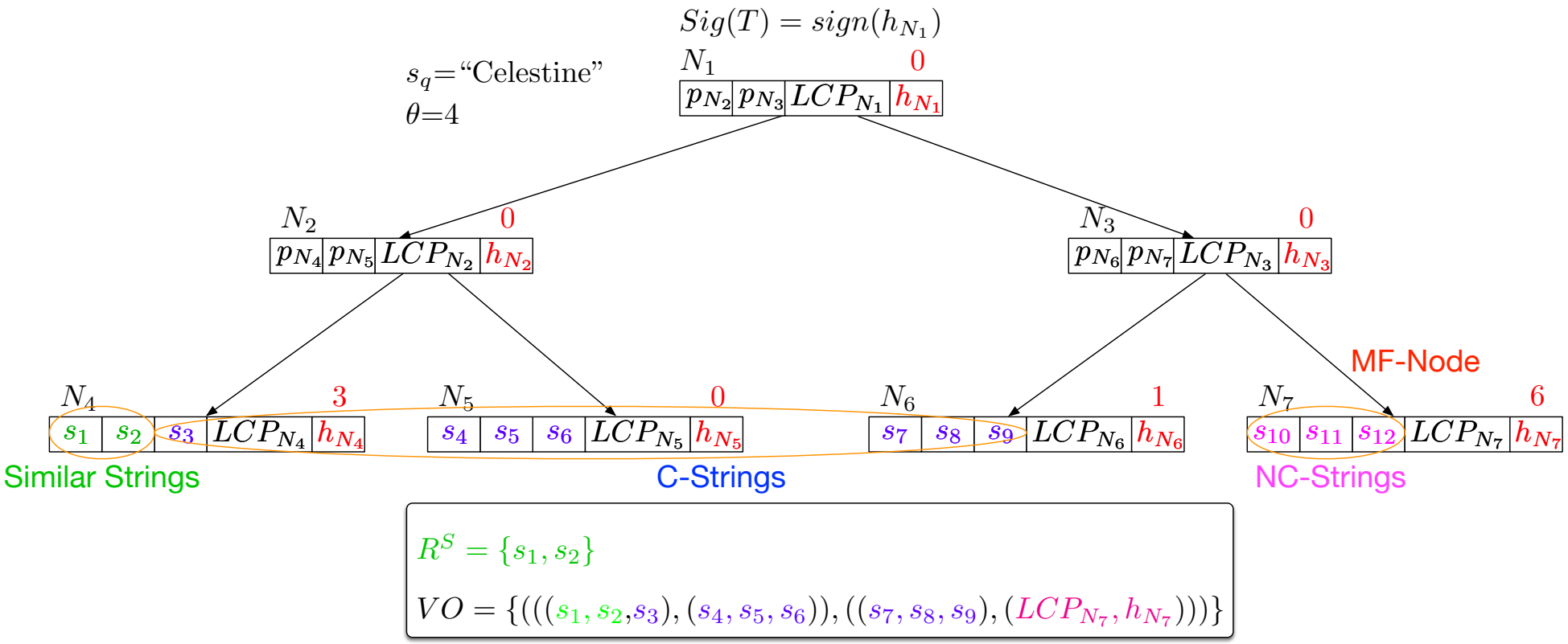
[DW16] propose an authenticated string indexing structure, named *MB*-tree.



- Store the longest common prefix (LCP) of the enclosed strings in every node.
- $\forall N$, calculate $MIN\_DST(s_q, N.LCP)$.
- If $MIN\_DST(s_q, N.LCP) > \theta$, then N is a MF-node.

ARM [DW16]

The server searches for the similar strings and constructs *VO* by traversing the *MB*-tree.



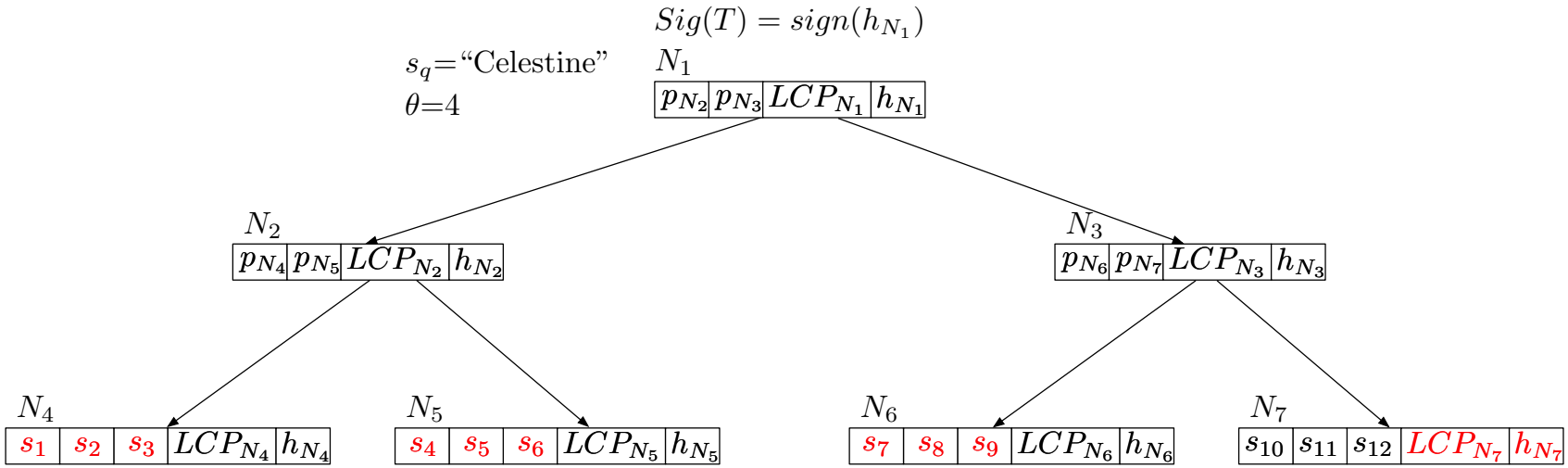
- Include all the C-strings and similar strings in *VO*.
- Substitute the large amount of NC-strings with the MF-nodes.

ARM [DW16]

The client checks the soundness of completeness of R^S by verifying the VO.

catches
- soundness violation
  - $\exists s \in R^S$, but $s \notin D$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$
- completeness violation
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$

$\boxed{\text{Compute } Sig(T) \text{ from } VO}$



$R^S = \{s_1, s_2\}$

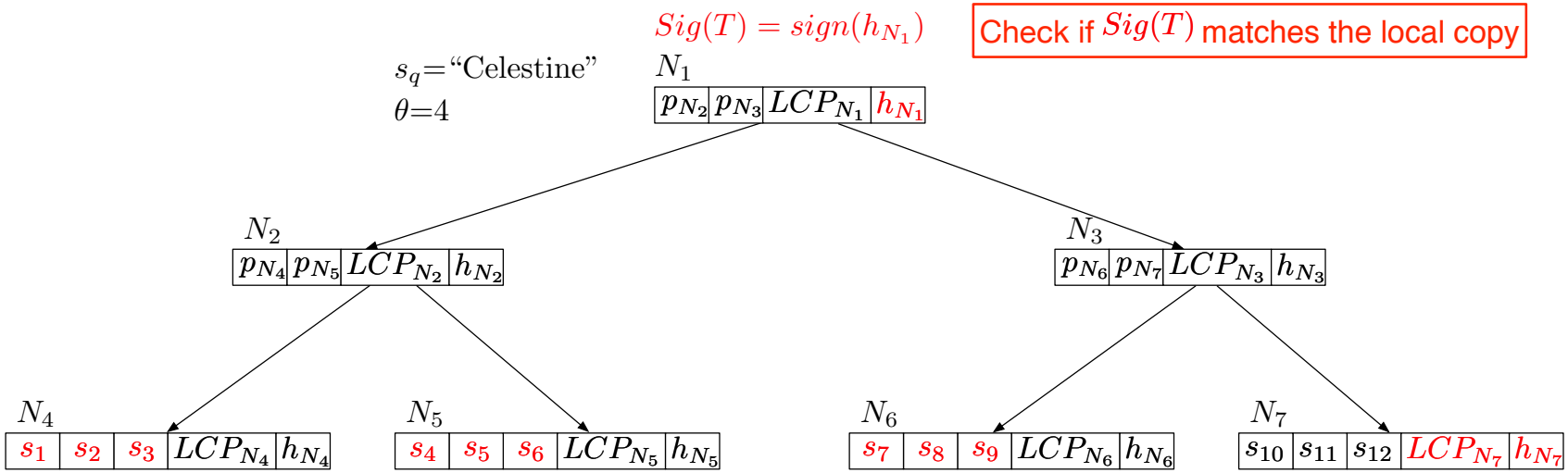
$VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}$

ARM [DW16]

The client checks the soundness and completeness of R^S by verifying the VO.

catches
- soundness violation
  - $\exists s \in R^S$, but $s \notin D$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$
- completeness violation
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$

Compute $Sig(T)$ from VO



$R^S = \{s_1, s_2\}$

$VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}$

ARM [DW16]

The client checks the soundness and completeness of R^S by verifying the *VO*.

catches
- soundness violation
  - $\exists s \in R^S$, but $s \notin D$ — $\boxed{\text{Compute } Sig(T) \text{ from } VO}$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$ — $\boxed{\forall s \in R^S, \text{ check if } DST(s, s_q) \leq \theta}$
- completeness violation
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$
    - $\boxed{\forall \text{C-string } s, \text{ check if } DST(s, s_q) > \theta}$
    - $\boxed{\forall \text{MF-node } N, \text{ check if } MIN\_DST(N.LCP, s_q) > \theta}$

$s_q =$ " Celestine "

$\theta = 4$

$R^S = \{s_1, s_2\}$

$VO = \{(((s_1, s_2, s_3), (s_4, s_5, s_6)), ((s_7, s_8, s_9), (LCP_{N_7}, h_{N_7})))\}$

for similar strings
$$\begin{cases} DST(s_1, s_q) = 4 \\ DST(s_2, s_q) = 3 < 4 \end{cases}$$

for C-strings
$$\begin{cases} DST(s_3, s_q) = 5 > 4 \\ DST(s_4, s_q) = 9 > 4 \\ DST(s_5, s_q) = 9 > 4 \\ DST(s_6, s_q) = 8 > 4 \\ DST(s_7, s_q) = 8 > 4 \\ DST(s_8, s_q) = 8 > 4 \\ DST(s_9, s_q) = 8 > 4 \end{cases}$$

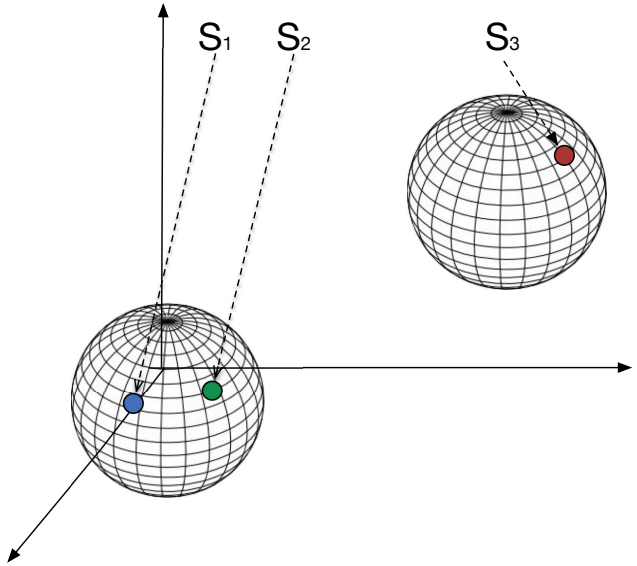
for MF-node
$$MIN\_DST(LCP_{N_7}, s_q) = 6 > 4$$

10 DST calculations

Problem: The verification cost is still high due to the large number of C-strings.

Embedding Function

Embedding maps strings into Euclidean points in a similarity-preserving way.

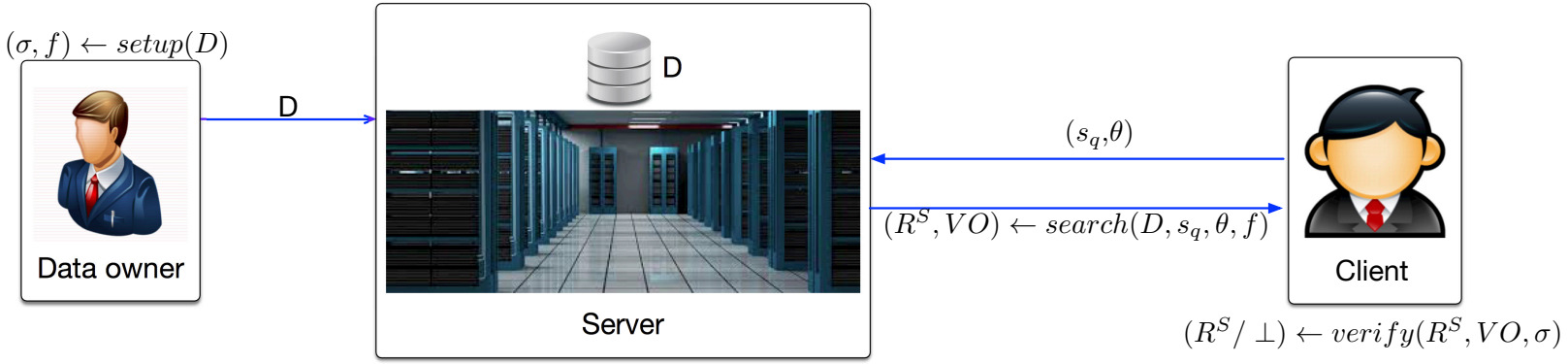


- Euclidean distance calculation is much more efficient than edit distance computing, i.e., $O(dst(p_i, p_j)) << O(DST(s_i, s_j))$.
- *SparseMap*[HS] is a *contractive* embedding approach, i.e., $dst(p_i, p_j) \leq DST(s_i, s_j)$.
- The complexity is $O(cn^2)$, where c is a small constant, and n is the number of strings.

Outline

Authentication Framework

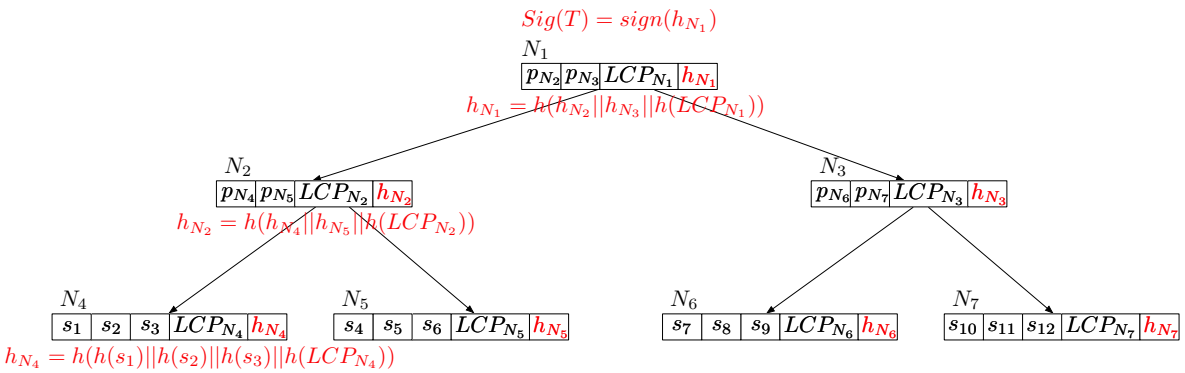
We require the server to construct *verification object (VO)* to demonstrate the soundness and completeness of the result.

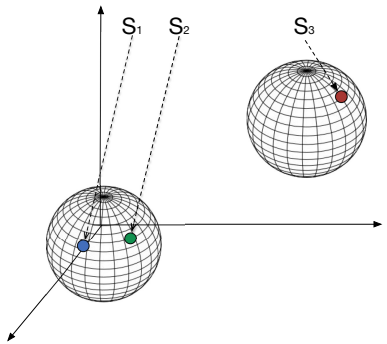


The client is able to efficiently detect any unsound or incomplete result returned by the server by checking the *VO*.

Authentication Preparation

- The data owner constructs the *MB*-tree.
- The data owner applies *SparseMap* to embed strings into Euclidean points.





Key idea For any C-string s, if $dst(p, p_q) > \theta$, it must be true that $DST(s, s_q) > \theta$.

VO Construction

Distant Bounding Hyper-rectangle (DBH) A hyper-rectangle R in the Euclidean space is a DBH if $min\_dst(p_q, R) > \theta$.

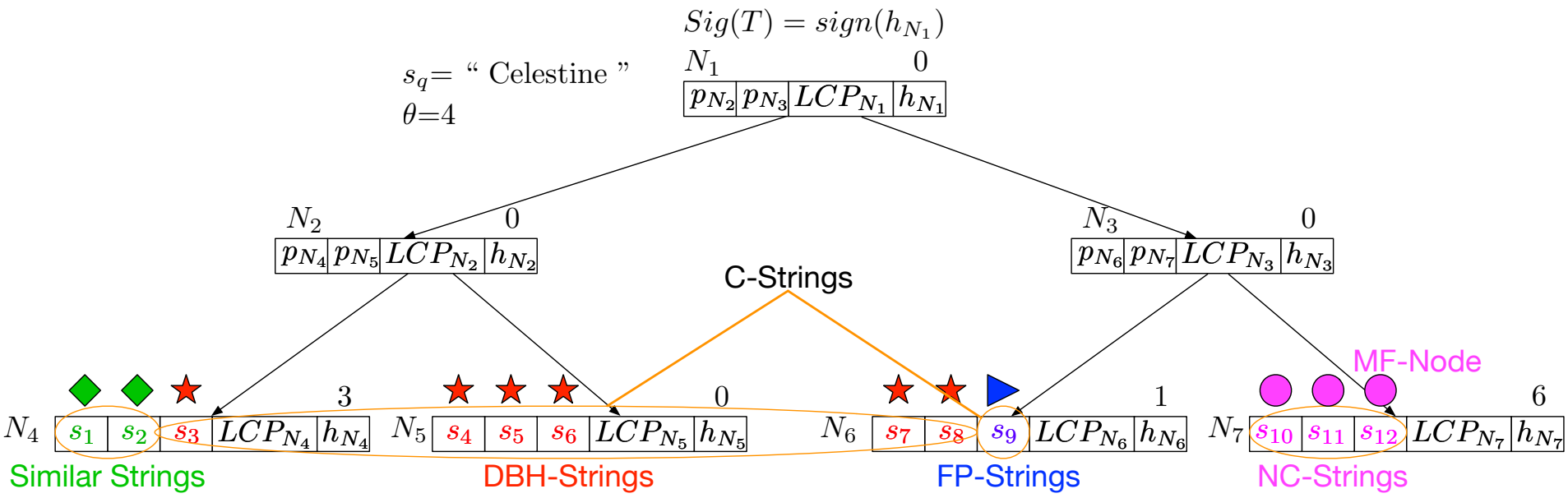
DBH-String For any C-string s, if $dst(p, p_q) > \theta$, we call it a DBH-string.

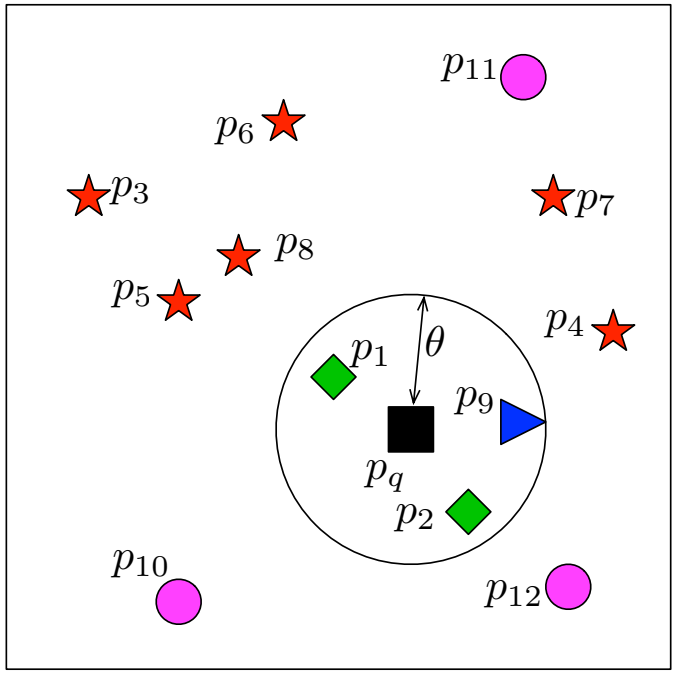
FP-String For any C-string s, if $dst(p, p_q) \leq \theta$, we call it a FP-string.

Key idea

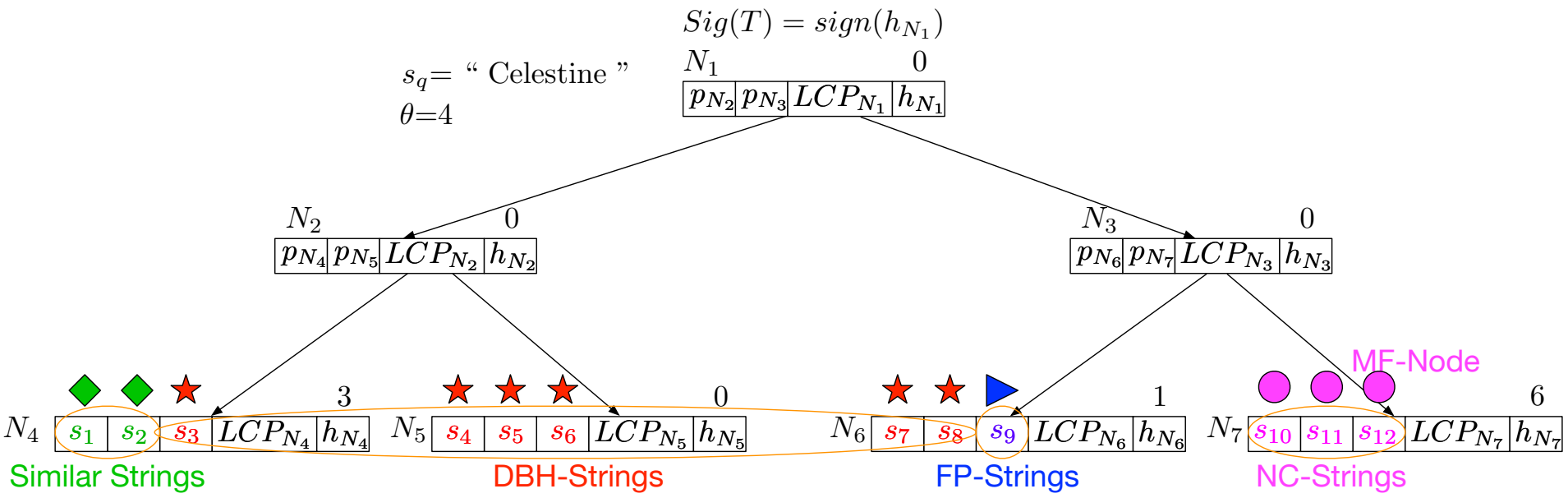
- To save the verification cost at the client side, the server should organize the set of DBH-strings into a small number of DBHs.
- By only checking the Euclidean distance between the target point p_q and the DBHs, the client assures that all DBH-strings are dis-similar to s_q.

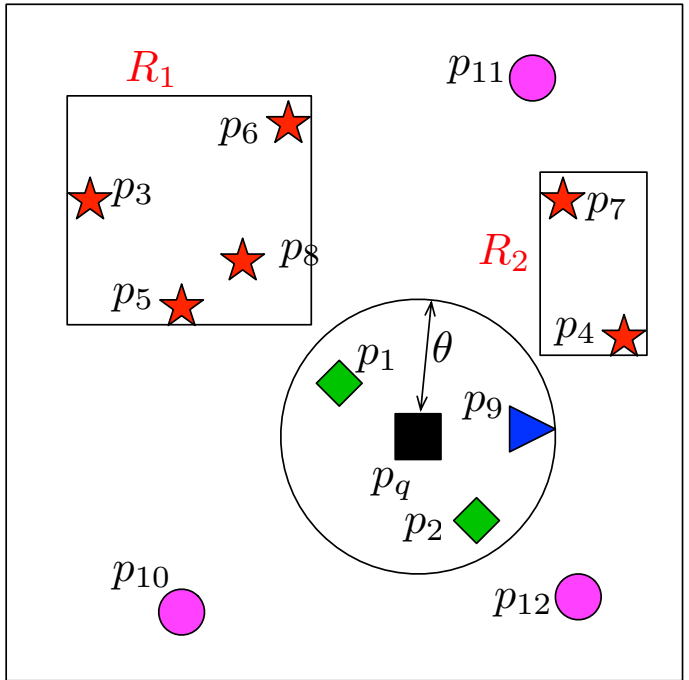
VO Construction





VO Construction





VO Construction

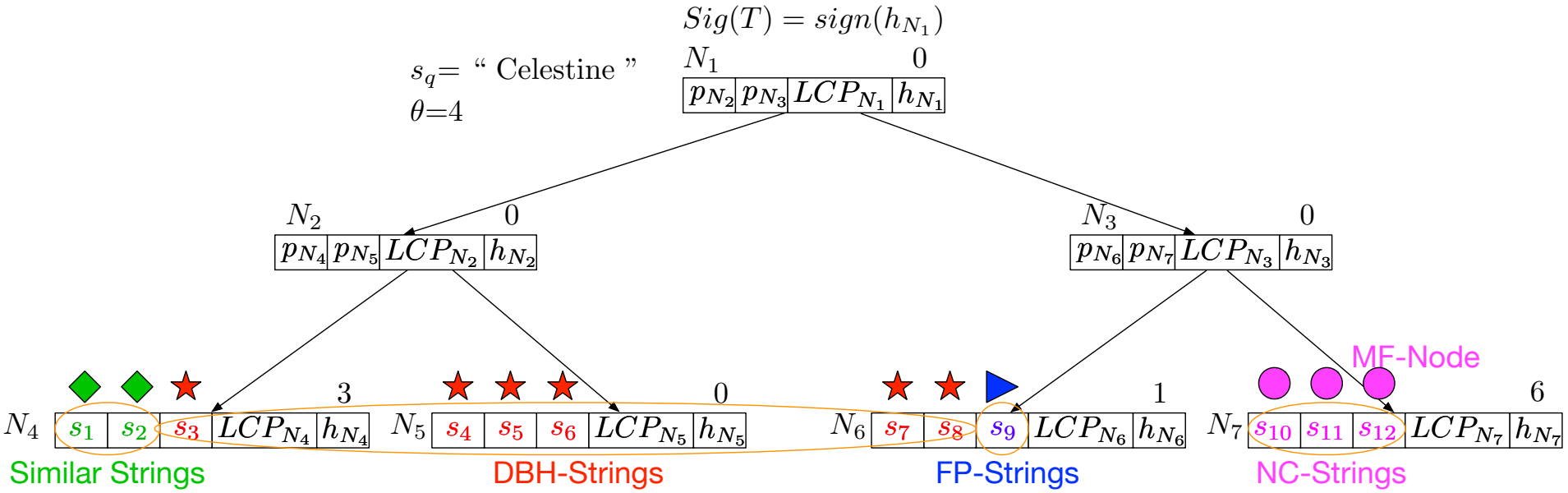
Theorem (NP-Completeness of DBH Construction)

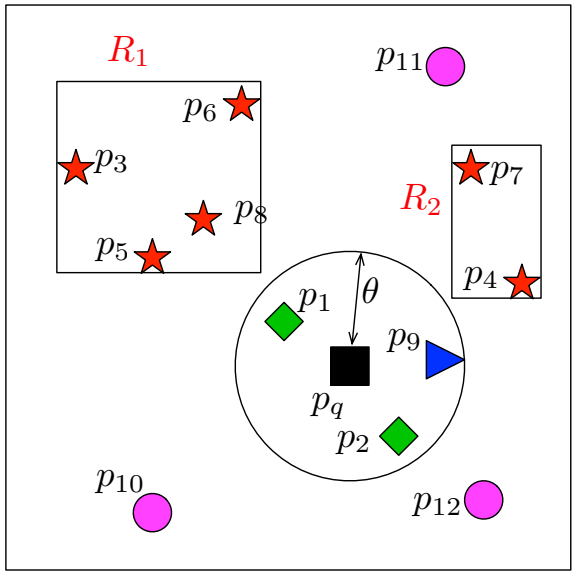
Given a query string s_q, and a set of DBH-strings $\{s_1, \ldots, s_t\}$, let $\{p_1, \ldots, p_t\}$ be their Euclidean points. It is a NP-complete problem to construct a mimimum number of rectangles $\mathcal{R} = \{R_1, \ldots, R_k\}$ s.t.
(1) $\forall i \neq j$, R_i and R_j do not overlap; and
(2) $\forall p_i$, there exists a R_j s.t. p_i is included in R_j.

- We design an efficient heuristic algorithm for the server to construct a small amount of *DBH*s.
- The complexity is cubic to the number of DBH-strings.

VO Construction

The server includes the *DBH*s in the *VO*.





$R^S = \{s_1, s_2\}$

$VO = \{(((s_1, s_2, (s_3, p_{R_1})), ((s_4, p_{R_2}), (s_5, p_{R_1}), (s_6, p_{R_1}))),$
$(((s_7, p_{R_2}), (s_8, p_{R_1}), s_9), (LCP_{N_7}, h_{N_7}))), \{R_1, R_2\}\}$

VO Verification

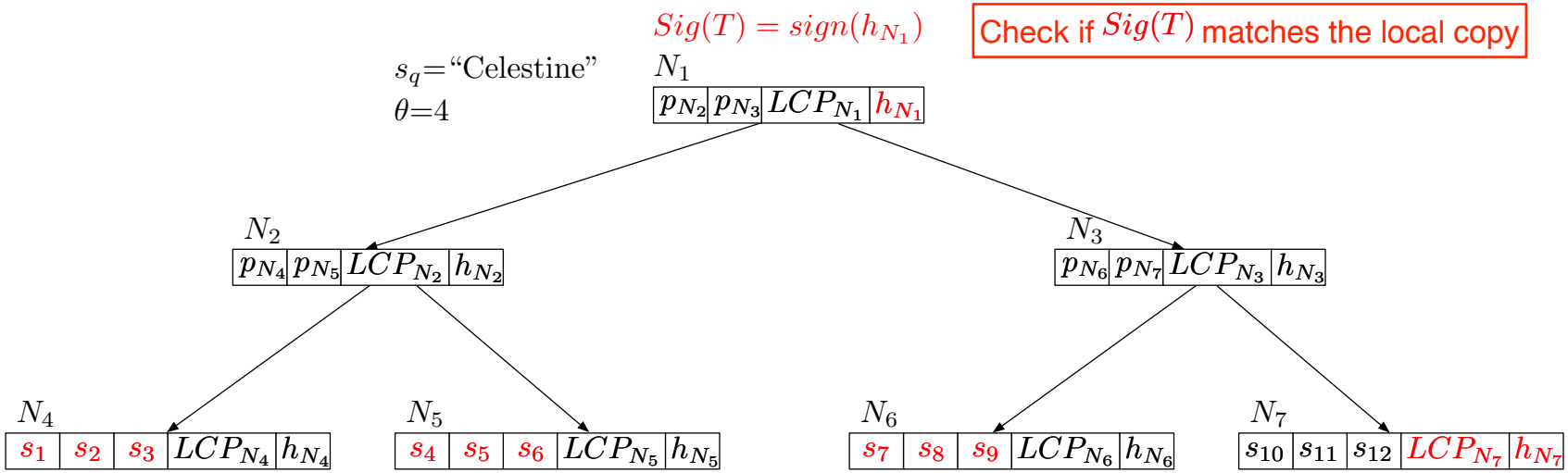
The client checks the soundness and completeness of R^S by verifying the *VO*.

Compute $Sig(T)$ from VO

catches:

- soundness violation:
  - $\exists s \in R^S$, but $s \notin D$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$
- completeness violation:
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$

Check if $Sig(T)$ matches the local copy



$R^S = \{s_1, s_2\}$

$$VO = \{(((s_1, s_2, (s_3, p_{R_1})), ((s_4, p_{R_2}), (s_5, p_{R_1}), (s_6, p_{R_1}))), (((s_7, p_{R_2}), (s_8, p_{R_1}), s_9), (LCP_{N_7}, h_{N_7}))), \{R_1, R_2\}\}$$

VO Verification

The client checks the soundness and completeness of R^S by verifying the *VO*.

catches:

- soundness violation
  - $\exists s \in R^S$, but $s \notin D$ — $\boxed{\text{Compute } Sig(T) \text{ from } VO}$
  - $\exists s \in R^S$, but $DST(s, s_q) > \theta$ — $\boxed{\forall s \in R^S, \text{ check if } DST(s, s_q) \leq \theta}$
- completeness violation
  - $\exists s \in D$ s.t. $DST(s, s_q) \leq \theta$ but $s \notin R^S$
    - $\boxed{\forall \text{MF-node } N, \text{ check if } MIN\_DST(N.LCP, s_q) > \theta}$
    - $\boxed{\forall \text{DBH-string } (s, p_R), \text{ check if } p \in R, \text{ and if } min\_dst(p_q, R) > \theta}$
    - $\boxed{\forall \text{FP-string } s, \text{ check if } DST(s_q, s) > \theta}$

$s_q =$ " Celestine "

$\theta = 4$

$R^S = \{s_1, s_2\}$

$$VO = \{(((s_1, s_2, (s_3, p_{R_1})), ((s_4, p_{R_2}), (s_5, p_{R_1}), (s_6, p_{R_1}))), (((s_7, p_{R_2}), (s_8, p_{R_1}), s_9), (LCP_{N_7}, h_{N_7}))), \{R_1, R_2\}\}$$

for similar strings $\begin{cases} DST(s_1, s_q) = 4 \\ DST(s_2, s_q) = 3 < 4 \end{cases}$

for MF-node $MIN\_DST(LCP_{N_7}, s_q) = 6 > 4$

for DBH-strings $\begin{cases} min\_dst(p_q, R_1) > \theta \\ min\_dst(p_q, R_2) > \theta \end{cases}$

for FP-string $DST(s_9, s_q) = 8 > 4$

4 DST calculations
+
2 dst calculations

ARM: 10 DST calculations

Complexity Analysis

Phase	Measurement	ARM [DW16]	EARRING (Our approach)
Setup	Time	$O(n)$	$O(cdn^2)$
	Space	$O(n)$	$O(n)$
VO Construction	Time	$O(n)$	$O(n + n_{DS}^3)$
	VO Size	$(n_R + n_C)\sigma_S + n_{MF}\sigma_M$	$(n_R + n_C)\sigma_S + n_{MF}\sigma_M + n_{DBH}\sigma_D$
VO Verification	Time	$O((n_R + n_{MF} + n_C)C_{Ed})$	$O((n_R + n_{MF} + n_{FP})C_{Ed} + n_{DBH}C_{El})$

(n: # of strings in D; c: a constant in [0, 1]; d: # of dimensions of Euclidean space; σ_S: the average length of the string; σ_M: Avg. size of a MB-tree node; σ_D: Avg. size of a DBH; n_R: # of strings in M^S; n_C: # of C-strings; n_{FP}: # of FP-strings; n_{DS}: # of DBH-strings; n_{DBH}: # of DBHs; n_{MF}: # of MF nodes; C_{Ed}: the complexity of an edit distance computation; C_{El}: the complexity of Euclidean distance calculation.)

- EARRING results in higher VO construction complexity at the server side.
- EARRING dramatically saves the VO verification cost at the client side.

Outline

Experiments

- Environment

  Language C++

  Testbed A Linux machine with 2.4 GHz CPU and 48 GB RAM

- Datasets

  Actors [1] $260,000$ lastnames

  Authors [2] $1,000,000$ full names

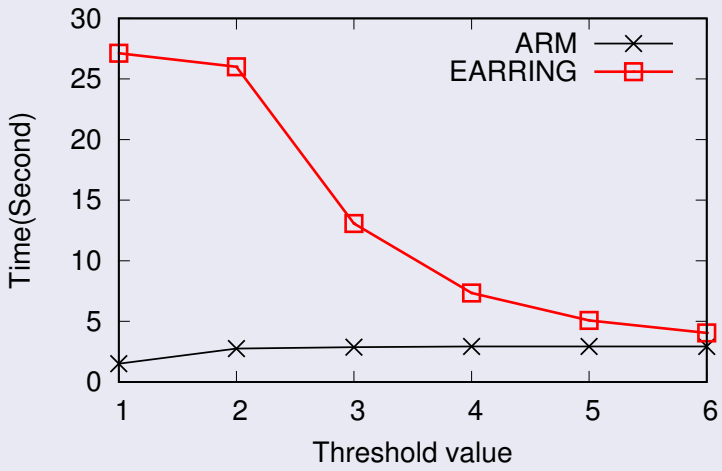
- Evaluation metric
  - VO construction time
  - VO verification time

[1] http://www.imdb.com/interfaces

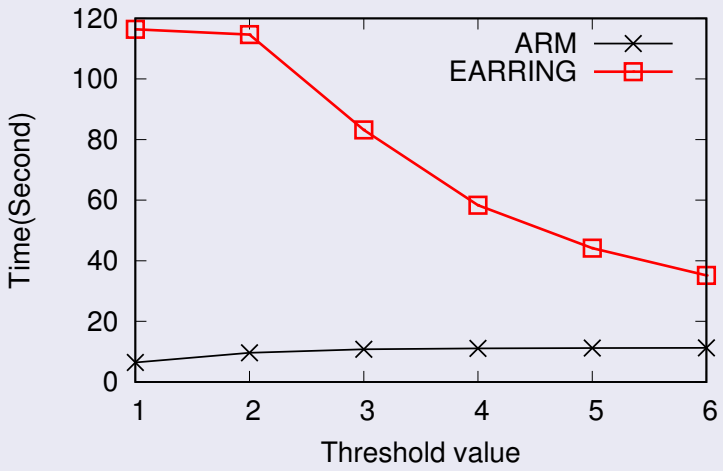
[2] http://dblp.uni-trier.de/xml/

VO Construction Time

Time Performance of VO Construction



(a) The *Actors* dataset

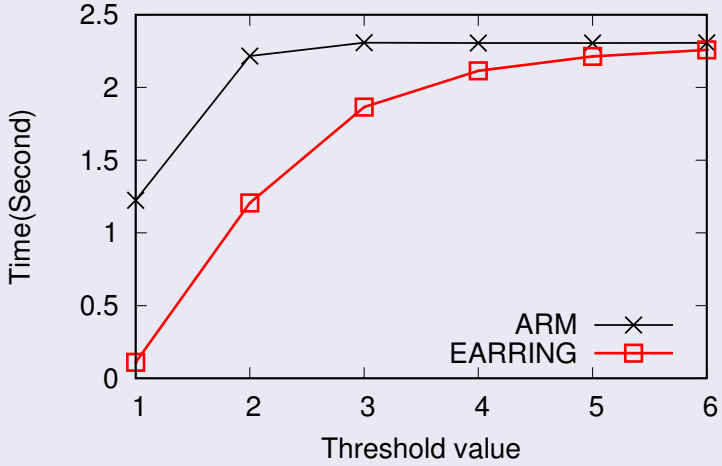


(b) The *Authors* dataset

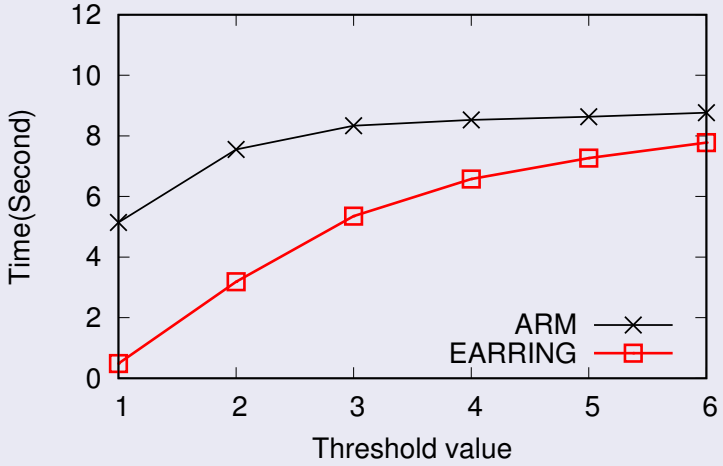
- EARRING takes more time at the server side to construct *VO*, especially when θ is small.

VO Verification Time

Time Performance of VO Verification



(a) The *Actors* dataset ($f = 1,000$)

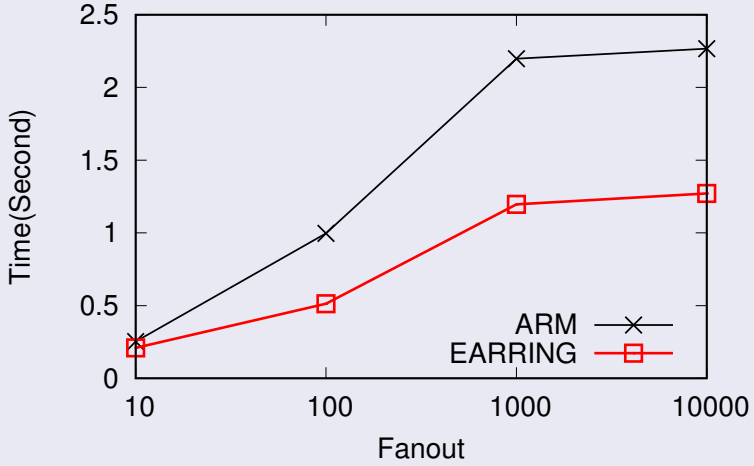


(b) The *Authors* dataset ($f = 1,000$)

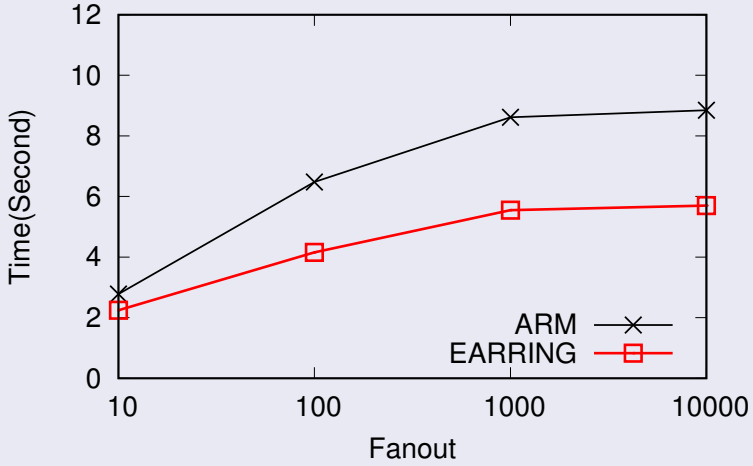
- EARRING is always more efficient than ARM [DW16] in terms of verification cost.
- The advantage of EARRING is large when θ is small.

Verification Time

Time Performance of VO Verification



(a) The *Actors* dataset ($f = 1,000$)



(b) The *Authors* dataset ($f = 1,000$)

- EARRING is always more efficient than ARM [DW16] in terms of verification cost.
- The advantage of EARRING is large when f is large.

Conclusion

- We design *EARRING*, an Efficient Authentication of outsouRced Record matchING.
- We prove that it is NP-complete to construct the minimum number of DBHs for the mismatching records, and design an efficient heuristic algorithm to build a small number of DBHs to represent the mismatching records.
- Experiment results demonstrate that *EARRING* saves up to 91% verification time at the client side compared with *ARM* [DW16].
- In the future, we plan to design the authentication methods that support other types of similarity metrics.

References I

[BS13] Sumeet Bajaj and Radu Sion.
Correctdb: Sql engine with practical query authentication.
VLDB Endowment, 2013.

[DLW14] Boxiang Dong, Ruilin Liu, and Wendy Hui Wang.
Prada: Privacy-preserving data-deduplication-as-a-service.
In *International Conference on Information and Knowledge Management*, 2014.

[DW16] Boxiang Dong and Wendy Wang.
Arm: Authenticated approximate record matching for outsourced databases.
In *International Conference on Information Reuse and Integration*, 2016.

[HS] G Hjaltason and H Samet.
Contractive embedding methods for similarity searching in metric spaces.
Technical report, Technical Report TR-4102, Computer Science Department.

[KV15] Dimitrios Karapiperis and Vassilios S Verykios.
An lsh-based blocking approach with a homomorphic matching technique for privacy-preserving record linkage.
IEEE Transactions on Knowledge and Data Engineering, 2015.

[MNT06] Einar Mykletun, Maithili Narasimha, and Gene Tsudik.
Authentication and integrity in outsourced databases.
ACM Transactions on Storage, 2006.

[NT05] Maithili Narasimha and Gene Tsudik.
Dsac: integrity for outsourced databases with signature aggregation and chaining.
In *International Conference on Information and Knowledge Management*, 2005.

References II

[YLY11] Man Lung Yiu, Eric Lo, and Duncan Yung.
Authentication of moving knn queries.
In *International Conference on Data Engineering*, 2011.

[YPPK08] Yin Yang, Stavros Papadopoulos, Dimitris Papadias, and George Kollios.
Spatial outsourcing for location-based services.
In *International Conference on Data Engineering*, 2008.

Q & A

Thank you!

Questions?