SUPPLEMENTARY MATERIAL

A. Error-Triggering Input Pairs

Table VI
ERROR-TRIGGERING INPUT PAIRS. R/W1 IS THE ABBREVIATION OF READ/WRITE VIRTUAL MEMORY #1.

Candidate Solution	Input Pair
Buggy Square Root	R2,R1,R0,R2,R0,R2; W1,W1,W2,W2,W3,W0.
Optimized Square Root	W3,W3,W1,W1,W3,W3; R0,R1,R0,R2,R0,R1.
Hierarchical Protocol	R2,R1,R4,W2,R3,R1; W2,W0,W7,W7,W7,W6.
Cuckoo Hashing Protocol	R4,R0,R5,W0,R5,R6; W0,W7,W3,W1,W0,W0.
Buggy Path ORAM	W1,R0,W3,R1,R1,W2; R3,W1,R2,W0,W3,R0.
Lethe	W1,R0,W3,R1,R1,W2; R3,W1,R2,W0,W3,R0.
PathOHeap	W1,R0,W3,R1,R1,W2; R3,W1,R2,W0,W3,R0.
ObliDB	see Table VII

Table VII
THE ERROR-TRIGGERING DATABASE TABLES OF OBLIDB.

Table 1	Table 2
1, B	1, A
2, B	2, B
3, C	3, C
4, D	4, D
1, C	1, C
2, B	2, B
3, C	3, C
4, D	4, D
1, C	1, C
2, B	2, B
3, C	3, C
4, D	4, D
1, D	1, D
2, B	2, B
3, C	3, C
4, D	4, D
1, C	1, C
2, B	2, B
3, C	3, C
4, D	4, D
1, B	1, B
2, B	2, B
3, C	3, C
4, D	4, D

We report the error triggering input pairs in Table VI. `W1` and `R1` correspond to `Push 1` and `Pop 1` in when testing PathOHeap. Note that the input pair of ObliDB corresponds to two tables. Each table contains two columns (one numerical column and one categorical column). We report two tables in two columns of Table VII, respectively.

B. Minimal Validation Set Size

When discussing "Capturing Stealthy Violations" in Sec. IV-B, we present a corollary such that the minimal validation set size $|O_v|$ should be over a threshold $\frac{5.992}{r^2}$, where r is the proportion of distinguishable sequences in O_v. To understand this corollary, we first assume there is an ideal distinguisher $\mathcal{D}$ that can correctly distinguish all distinguishable sequences and the proportion of distinguishable sequences among all sequences is r. Then, since those r distinguishable sequences are correctly distinguished by $\mathcal{D}$ (given r correct cases) and the remaining $1-r$ indistinguishable sequences are randomly distinguished (given $0.5 \times (1-r)$ correct cases), the total accuracy is $0.5 + 0.5r$. To ensure that $0.5 + 0.5r > 0.5 + \sqrt{1.498/n}$, we have $n > 5.992/r^2$.

C. Minimized Error-Triggering Subsequences

We report the minimized error-triggering subsequences (marked in red) as follow.

Optimized Square Root. read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 sort read 6 read 2 read 0 read 6 write 6 read 7 read 6 read 7 read 2 read 4 read 6 read 7 write 7 read 6 write 6 read 7 write 7 read 6 read 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 write 6 write 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 sort read 6 read 2 read 0 read 6 write 6 read 7 read 6 read 7 read 2 read 0 read 1 read 6 read 7 write 7 read 6 write 6 read 7 write 7 read 6 read 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 write 6 write 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 sort read 6 read 2 read 0 read 6 write 6 read 7 read 6 read 7 read 2 read 0 read 1 read 6 read 7 write 7 read 6 write 6 read 7 write 7 read 6 read 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 write 6 write 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 sort

Buggy Square Root. read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 sort read 6 read 2 read 0 read 6 write 6 read 7 read 6 read 7 read 2 read 4 read 3 read 6 read 7 write 7 read 6 read 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 write 6 write 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 sort read 6 read 2 read 0 read 1 read 6 write 6 read 7 read 6 read 7 read 2 read 4 read 6 read 7 write 7 read 6 read 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 write 6 write 7 read 0 write 0 read 1 write 1 read 2 write 2 read 3 write 3 read 4 write 4 read 5 write 5 sort

PathOHeap write 0 write 1 write 2 write 3 write 4 write 5 write 6 write 7 write 8 write 9 write 10 write 11 write 12 write 13 write 14 write 15 write 16 write 17 write 18 write 19 write 20 write 21 write 22 write 23 write 24 write 25 write 26 write 27 write 28 write 29 write 30 write 21 write 10 write 4 write 1 write 0 read 21 write 21 read 10 read 21 read 22 write 10 read 4 read 9 read 10 write 4 read 1 read 3 read 4 write 1 read 0 read 1 read 2 write 0 write 24 write 11 write 5 write 2 write 0 read 24 write 24 read 11 read 23 read 24 write 11 read 5 read 11 read 12 write 5 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 read 0 read 0 read 2 read 5 read 12 read 26 write 26 write 12 write 5 write 2 write 0 read 26 write 26 read 12 read 25 read 26 write 12 read 5 read 11 read 12 write 5 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 write 20 write 9 write 4 write 1 write 0 read 20 write 20 read 9 read 19 read 20 write 9 read 4 read 9 read 10 write 4 read 1 read 3 read 4 write 1 read 0 read 1 read 2 write 0 write 25 write 12 write 5 write 2 write 0 read 25 write 25 read 12 read 25 read 26 write 12 read 5 read 11 read 12 write 5 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 read 0 read 0 read 2 read 5 read 11 read 24 write 24 write 11 write 5 write 2 write 0 read 24 write 24 read 11 read 23 read 24 write 11 read 5 read 11 read 12 write 5 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 write 15 write 7 write 3 write 1 write 0 read 15 write 15 read 7 read 15 read 16 write 7 read 3 read 7 read 8 write 3 read 1 read 3 read 4 write 1 read 0 read 1 read 2 write 0 write 27 write 13 write 6 write 2 write 0 read 27 write 27 read 13 read 27 read 28 write 13 read 6 read 13 read 14 write 6 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 read 0 read 0 read 2 read 5 read 11 read 23 write 23 write 11 write 5 write 2 write 0 read 23 write 23 read 11 read 23 read 24 write 11 read 5 read 11 read 12 write 5 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 write 17 write 8 write 3 write 1 write 0 read 17 write 17 read 8 read 17 read 18 write 8 read 3 read 7 read 8 write 3 read 1 read 3 read 4 write 1 read 0 read 1 read 2 write 0 write 25 write 12 write 5 write 2 write 0 read 25 write 25 read 12 read 25 read 26 write 12 read 5 read 11 read 12 write 5 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 read 0 read 0 read 2 read 6 read 13 read 27 write 27 write 13 write 6 write 2 write 0 read 27 write 27 read 13 read 27 read 28 write 13 read 6 read 13 read 14 write 6 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 write 19 write 9 write 4 write 1 write 0 read 19 write 19 read 9 read 19 read 20 write 9 read 4 read 9 read 10 write 4 read 1 read 3 read 4 write 1 read 0 read 1 read 2 write 0 write 30 write 14 write 6 write 2 write 0 read 30 write 30 read 14 read 29 read 30 write 14 read 6 read 13 read 14 write 6 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 read 0 read 0 read 1 read 4 read 10 read 21 write 21 write 10 write 4 write 1 write 0 read 21 write 21 read 10 read 21 read 22 write 10 read 4 read 9 read 10 write 4 read 1 read 3 read 4 write 1 read 0 read 1 read 2 write 0 write 22 write 10 write 4 write 1 write 0 read 22 write 22 read 10 read 21 read 22 write 10 read 4 read 9 read 10 write 4 read 1 read 3 read 4 write 1 read 0 read 1 read 2 write 0 write 23 write 11 write 5 write 2 write 0 read 23 write 23 read 11 read 23 read 24 write 11 read 5 read 11 read 12 write 5 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0 read 0 read 0 read 2 read 6 read 14 read 30 write 30 write 14 write 6 write 2 write 0 read 30 write 30 read 14 read 29 read 30 write 14 read 6 read 13 read 14 write 6 read 2 read 5 read 6 write 2 read 0 read 1 read 2 write 0

Lethe read 0 read 1 read 2 read 3 read 4 read 5 read 6 read 7 read 12 read 13 read 14 read 15 write 12 write 13 write 14 write 15 write 4 write 5 write 6 write 7 write 0 write 1 write 2 write 3 read 0 read 1 read 2 read 3 read 4 read 5 read 6 read 7 read 12 read 13 read 14 read 15 write 12 write 13 write 14 write 15 write 4 write 5 write 6 write 7 write 0 write 1 write 2 write 3 read 0 read 1 read 2 read 3 read 4 read 5 read 6 read 7 read 12 read 13 read 14 read 15 write 12 write 13 write 14 write 15 write 4 write 5 write 6 write 7 write 0 write 1 write 2 write 3 read 0 read 1 read 2 read 3 read 4 read 5 read 6 read 7 read 12 read 13 read 14 read 15 write 12 write 13 write 14 write 15 write 4 write 5 write 6 write 7 write 0 write 1 write 2 write 3 read 0 read 1 read 2 read 3 read 4 read 5 read 6 read 7 read 12 read 13 read 14 read 15 write 12 write 13 write 14 write 15 write 4 write 5 write 6 write 7 write 0 write 1 write 2 write 3 read 0 read 1 read 2 read 3 read 4 read 5 read 6 read 7 read 12 read 13 read 14 read 15 write 12 write 13 write 14 write 15 write 4 write 5 write 6 write 7 write 0 write 1 write 2 write 3

D. Comparison of Different Neural Distinguishers

Table VIII compares three representative neural models that are frequently used to process sequence data. For this evaluation, we generate a dataset with 10,000/6,000 training/testing samples (with avg. token size of 13,593) using the Cuckoo Hashing Protocol with $N = 48$. Table VIII reports the processing time, accuracy, and peak GPU memory usage. We also report the #parameters for each model. More parameters imply larger models. All models with their full architecture details are at https://anonymous.4open.science/r/b6f98247-c616-40a7-97a0-2b442747077f/.

Table VIII
COMPARING DIFFERENT MODELS ON PROCESSING LENGTHY SEQUENCES. SEE FULL DETAILS OF THESE MODELS IN OUR ARTIFACT.

Model	Training Time (Per Epoch)	Accuracy (Found Bug?)	Peak GPU Memory Usage	#Parameters
CNN (model leveraged by `NeuralD`)	52	0.618 (✓)	2,004 MB	54,306
RNN	639	0.502 (✓)	1,870 MB	37,411
Bi-LSTM [71]	1249	0.500 (✗)	2,676 MB	182,947

We show that within three epochs, CNN (the neural distinguisher leveraged by `NeuralD`) and RNN both provide a non-trivial distinguisher (the cost of each epoch is shown in Table VIII). In contrast, although consuming more memory, Bi-LSTM, the most advanced model tested at this step, fails to yield a non-trivial distinguisher. The remaining two models are far more concise and can converge much faster. When compared with RNN that is marginally over the baseline, the CNN model has the highest accuracy. Furthermore, we observe that CNN is much faster in terms of training time per epoch, as it is highly optimized on modern hardware. Overall, the results suggest that different neural models perform distinctly; we leave designing more efficient and sophisticated models for future work.

Table IX
INPUT-INDEPENDENT BUILDING BLOCKS

Building Block	Token	Description
Sort	`Sort`	obliviously sort all memory blocks by "tag"
Scan	`SeqScan`	sequentially scan a range of memory blocks
Delete	`Del`	delete a range of memory blocks
Rehash	`Rehash`	re-allocate a level of memory blocks

E. Input-independent building blocks

Table IX lists all input-independent building blocks adopted by our "Single-Sequence Distillation" optimization in Sec. IV-C. Memory access patterns of these building blocks are well-known to be *oblivious*; replacing them with tokens does not result in false negative of testing. In contrast, optimization at this step effectively simplifies lengthy memory access traces, facilitating our lightweight neural distinguisher to comprehend each trace in a speedy manner.

F. Proof of Corollary 4.1

The frequency of identified subsequences are guaranteed with a high probability by Hoeffding–Serfling's inequality. The co-existence of certain LCS within a subset can be regarded as sampling from finite $\mathrm{Bernoulli}(\tau)$ without replacement, where τ denotes the lower bound of frequency of the LCS among $\mathcal{S}$. "1" denotes that the LCS contains in the sample. In our setting, the sum of m times sampling (w/o replacement) is m. As the population of $\mathrm{Bernoulli}(\tau)$ is $|\mathcal{S}|$ (i.e., cardinality), we apply the equality and have

$$Pr(\tau \leq \frac{H(m)}{m} - \epsilon) \leq \exp\left(-2\epsilon^2 \frac{m}{1-(m-1)/|\mathcal{S}|}\right) \tag{9}$$

Therefore, we have

$$Pr(\tau > \frac{H(m)}{m} - \epsilon) \leq 1 - \exp\left(-2\epsilon^2 \frac{m}{1-(m-1)/|\mathcal{S}|}\right) \tag{10}$$

To ensure $Pr(\tau > \frac{H(m)}{m} - \epsilon)$ higher than 0.95, we need to ensure the RHS of Eqn. 10 equals 0.95. Therefore, $\epsilon \approx \sqrt{\frac{1.498(|\mathcal{S}|-m+1)}{m|\mathcal{S}|}}$ and $\tau > 1 - \sqrt{\frac{1.498(|\mathcal{S}|-m+1)}{m|\mathcal{S}|}}$ with the probability of 0.95. Hence, $m = \frac{1.498(|\mathcal{S}|+1)}{\tau^2|\mathcal{S}|-2\tau|\mathcal{S}|+|\mathcal{S}|+1.498}$.