

Tuning and Attacking Secure Learned Bloom Filters

Calvin Higgins

Department of Computer Science and Statistics
University of Rhode Island

April 22, 2025

Approximate Membership Query (AMQ) Filters

Idea: Check if $x \in S$ with false positive rate ϵ .

Approximate Membership Query (AMQ) Filters

Motivation: Cheaply filter database queries to reduce load.

Bloom Filters (BFs) [**bloom** 1970]

Idea: Output $x \in S$ iff x maps to all ones.

Bloom Filters (BFs) [**bloom** 1970]

Parameters: The number of hash functions and bits.

Learned Bloom Filters (LBFs)

[kraska`beutel`chi`dean`polyzotis`2018]

Idea: Try machine learning model and then eliminate false negatives.

Learned Bloom Filters (LBFs)

[kraska`beutel`chi`dean`polyzotis`2018]

Parameters: The threshold and Bloom filter false positive rate.

Adversarial Objective

Create **new** false positives to reduce database performance.

Adversarial Capabilities

The adversary has

- 1 Polynomial time
- 2 Query access to the AMQ filter
- 3 White-box access to the AMQ filter
- 4 Total control over the set S

Secure Bloom Filters (SBFs) [naor·eylon·2019]

Idea: Replace hash functions with cryptographic hash functions.

Secure Learned Bloom Filters (SLBFs) [bishop·tirmazi·2025]

Idea: Double-check machine learning model with secure Bloom filters.

Secure Learned Bloom Filters (SLBFs) [bishop·tirmazi·2025]

Parameters: The threshold and secure Bloom filter false positive rates.

Tuning Secure Learned Bloom Filters (SLBFs)

False Positive Rate [bishop'tirmazi'2025]

SBLFs have an expected false positive rate of

$$\epsilon = M_{\text{FPR}} \text{TP}_{\text{FPR}} + M_{\text{TNR}} \text{FN}_{\text{FPR}}$$

where M is the machine learning model, TP is the true positive filter and FN is the false negative filter.

Memory Footprint

SLBFs require

$$m = -\frac{n}{\ln(2)^2} (M_{\text{TPR}} \ln(\text{TP}_{\text{FPR}}) + M_{\text{FNR}} \ln(\text{FN}_{\text{FPR}})) + M_m + 2\lambda$$

bits of memory.

Tuning Secure Learned Bloom Filters (SLBFs)

Optimal Secure Bloom Filter False Positive Rates

The optimal true positive and false negative filter false positive rates are

$$TP_{FPR}^* = \epsilon \frac{M_{TPR}}{M_{FPR}} \qquad FN_{FPR}^* = \epsilon \frac{M_{FNR}}{M_{TNR}}$$

where M is the machine learning model, TP is the true positive filter and FN is the false negative filter.

Proof Sketch

Express FN_{FPR} in terms of TP_{FPR} . Set $\frac{\partial m}{\partial TP_{FPR}} = 0$ and solve for TP_{FPR}^* .
Substitute to find FN_{FPR}^* .

Tuning Secure Learned Bloom Filters (SLBFs)

Optimal Threshold

The optimal threshold τ^* minimizes

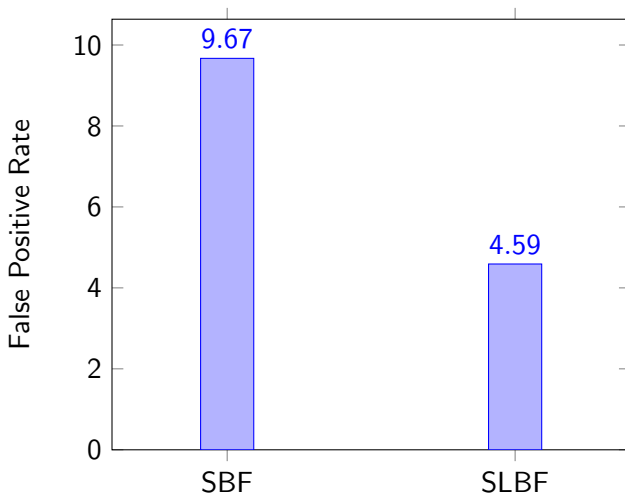
$$C(\tau) = -M_{\text{TPR}} \ln \left(\frac{M_{\text{TPR}}}{M_{\text{FPR}}} \right) - M_{\text{FNR}} \ln \left(\frac{M_{\text{FNR}}}{M_{\text{TNR}}} \right)$$

subject to

$$\max \left\{ \epsilon \frac{M_{\text{TPR}}}{M_{\text{FPR}}}, \epsilon \frac{M_{\text{FNR}}}{M_{\text{TNR}}} \right\} \leq \epsilon_{\max}$$

and can be found in $\Theta(n \lg n)$ time with dynamic programming!

Tuning Secure Learned Bloom Filters (SLBFs)



SLBFs outperform SBFs under a fixed memory budget!

Attacking Secure Learned Bloom Filters (SLBFs)

Experimental Setup:

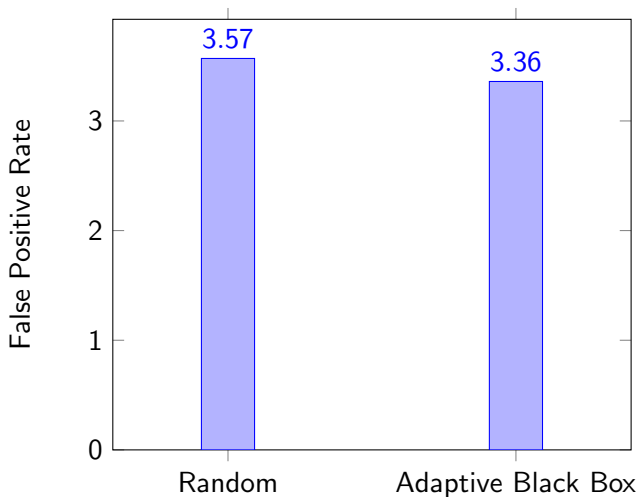
- 1 Truncate/pad $\sim 600\text{K}$ benign/malicious URLs to 128 bytes.
- 2 Train a shallow Transformer encoder for URL classification.
- 3 Compute optimal SLBF parameter values.
- 4 Construct a SLBF ($\epsilon = 0.05$, $\epsilon_{\max} = 0.2$) from the set of malicious URLs.

Attacking Secure Learned Bloom Filters (SLBFs)

Adaptive Black Box Attack:

- ❶ Create a synthetic SLBF
 - ❶ Generate a random set of URLs.
 - ❷ Label the URLs with the SLBF.
 - ❸ Train a deep Transformer encoder on the labeled URLs.
- ❷ Create adversarial examples
 - ❶ Generate a new random set of URLs.
 - ❷ Embed the URLs with the embedding layer.
 - ❸ Optimize the URL embeddings with Adam to have positive class label.
 - ❹ Unembed the URL embeddings with maximum cosine similarity.
 - ❺ Remove any previously queried URLs with an AMQ filter.
 - ❻ Return the optimized URLs.

Attacking Secure Learned Bloom Filters (SLBFs)



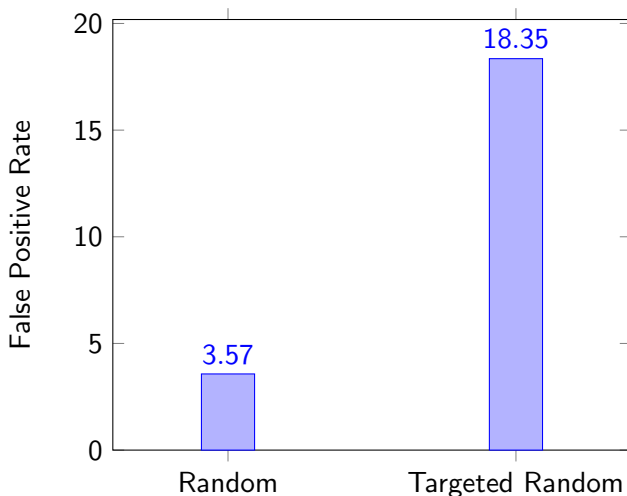
No significant difference! ($p = 0.1532$, $\chi^2 = 266$)

Attacking Secure Learned Bloom Filters (SLBFs)

Targeted Random Attack:

- 1 Generate a random set of URLs.
- 2 Label the random URLs with the learned model.
- 3 Split the URLs into two sets by label.
- 4 If the true positive filter has higher false positive rate, take the positive URL set, otherwise take the negative URL set.
- 5 Remove any previously queried URLs with an AMQ filter.
- 6 Return the optimized URLs.

Attacking Secure Learned Bloom Filters (SLBFs)



Significant but the defense succeeded ($\epsilon_{\max} = 0.2$)!

Contributions:

- ➊ Implemented Secure Bloom filters
- ➋ Implemented Secure Learned Bloom filters
- ➌ Derived optimal parameters for Secure Learned Bloom filters
 - ➊ Closed-form solutions for FN_{FPR} and TP_{FPR}
 - ➋ Efficient dynamic programming algorithm for τ
- ➍ Evaluated the robustness of Secure Learned Bloom filters against
 - ➊ Adaptive black box attack
 - ➋ Targeted randomized attack

References