Adversarially Robust Bloom Filters: Privacy, Reductions, and Open Problems

Hayder Tirmazi

City College of New York, CUNY, New York, USA

Abstract. A Bloom filter is a space-efficient probabilistic data structure that represents a set S of elements from a larger universe U. This efficiency comes with a trade-off, namely, it allows for a small chance of false positives. When you query the Bloom filter about an element x, the filter will respond 'Yes' if $x \in S$. If $x \notin S$, it may still respond 'Yes' with probability at most ε. We investigate the adversarial robustness and privacy of Bloom filters, addressing open problems across three prominent frameworks: the game-based model of Naor-Oved-Yogev (NOY), the simulator-based model of Filić et. al., and learning-augmented variants. We prove the first formal connection between the Filić and NOY models, showing that Filić correctness implies AB-test resilience. We resolve a longstanding open question by proving that PRF-backed Bloom filters fail the NOY model's stronger BP-test. Finally, we introduce the first private Bloom filters with differential privacy guarantees, including constructions applicable to learned Bloom filters. Our taxonomy organizes the space of robustness and privacy guarantees, clarifying relationships between models and constructions.

Keywords: Bloom filters · pseudorandomness · differential privacy

1 Introduction

A Bloom filter is a probabilistic data structure that encodes a set S from some large but finite universe U. Bloom filters are used to answer membership queries, i.e., for some $x \in U$, is $x \in S$? Bloom filters use less memory than explicitly encoding S, but at the cost of false positives. For any x if $x \in S$, the Bloom filter will return true with probability 1. If $x \notin S$, the Bloom filter might still incorrectly return true with probability at most ε for some $\varepsilon \in [0, 1]$. Bloom filters are widely deployed in critical real-world systems such as Google's LevelDB [Goo23], Meta's RocksDB [Met], and the Linux Kernel [Fou23]. This has made the adversarial robustness of Bloom filters a growing concern [GKL14, NE19].

Bloom filters have historically only been analyzed in a non-adversarial setting where the false positive probability of an element x uniformly randomly chosen from U is computed over the internal randomness of the Bloom filter construction [BM04]. A series of recent works has focused, instead, on the performance of Bloom filters in the presence of adversaries. Naor et al. [NE19, NO22, LN25] propose game-based robustness notions such as the Always-Bet (AB) and Bet-or-Pass (BP) tests for Bloom filters. Almashaqbeh et al. [ABT24] extend these game-based notions to learned Bloom filters, which are a variant of Bloom filters that use machine learning models. Filić et al. develop simulator-based robustness notions. Despite significant recent progress, the relationships between these models remain unclear, and several important problems remain unanswered.

The goal of this work is to unify and advance this recently developed theory of Bloom filter adversarial robustness. We articulate 10 open problems that span adversarial models,

E-mail: hayder.research@gmail.com (Hayder Tirmazi)

construction styles, and privacy goals. We contribute to 3 of these problems and leave the remaining 7 problems as open directions for research in this area. In terms of our discussed open problems (Section 3), we partially solve Problem 1, and mostly solve Problems 5 and 6.

Private Bloom filters: We introduce the first Bloom filter constructions that satisfy differential privacy guarantees. In particular, we introduce two constructions, the Mangat filter and the Warner filter, based on well-known randomized response mechanisms. We also fill a gap in this area by showing that Mangat's randomized response satisfies the notion of asymmetric differential privacy.

Bridging NOY model and Filić model: We show that Filić correctness implies AB-test resilience, marking the first known formal connection between these security frameworks. We also show that AB-test and BP-test resilience do not imply Filić correctness.

PRF-backed Standard Bloom filter: We prove that, in all practical cases, a PRF-backed Standard Bloom filter does not satisfy the NOY model's notion of BP-test resilience. This was left as an open question by Naor and Oved [NO22].

After covering related work, the remainder of this paper is organized as follows. We discuss preliminaries in Section 2. Section 3 is dedicated to discussing the open problems we enumerate in this work and providing a taxonomy of them. Section 4 covers the private Bloom filter constructions we introduce in this work. Section 5 introduces formal connections between the NOY model and the Filić model. Section 6 proves the result regarding PRF-backed Standard Bloom filters not being resilient under the BP-test.

1.1 Related Work

Gerbet et al. [GKL14] suggests practical attacks on Bloom filters and the use of universal hash functions and MACs to mitigate a subset of those attacks. Naor and Yogev [NE19] define an adversarial model for Bloom filters and provide a method for constructing adversary-resilient Bloom filters. Naor and Oved [NO22] present several robustness notions in a generalized adversarial model for Bloom filters. Clayton et al. [CPS19] and Filić et al. [FPUV22] provide secure constructions for Bloom filters using a game-based and a simulator-based model, respectively. Reviriego et al. [RHDS21] propose a practical attack on learned Bloom filters. They suggest possible mitigations, e.g., swapping to a classical Bloom filter upon attack detection. Almashaqbeh et al. [ABT24] propose provably secure learned Bloom filter constructions by extending the adversarial model of Naor et al. [NE19, NO22].

Many works, including Sengupta et al. [SBBR17], Reviriego et al. [RSMW+22], and Galan et al. [GRW+22] have shown that Bloom filters are vulnerable to set reconstruction attacks, i.e., given the internal state of a Bloom filter it is possible to infer the set the Bloom filter stores with high probability. Bianchi et al. [BBL12] provide privacy metrics for Bloom filters. Bianchi et al.'s metrics are based on k-anonymity [Swe02]. Filić et al. [FPUV22] propose a simulator-based notion of privacy for Bloom filters based on information leakage profiles. They provide privacy bounds for Bloom filters that use pseudo-random functions on their input set. Filić et al.'s proposal does not achieve meaningful privacy for Bloom filters whose input sets have low min-entropy, and their notion of Elem-Rep privacy is not immune to set reconstruction attacks from computationally unbounded adversaries. Concurrently and independently of our work, Ke et al. [KLS+25] propose a differentially private Bloom filter construction in a preprint dated February 2, 2025. Our Warner filter, introduced in the first version of this preprint publicly available on January 27, 2025, provides a similar differential privacy guarantee using Warner's randomized response mechanism. To the best of our knowledge, ours is the earliest work to formally analyze

differential privacy for Bloom filters in this setting. We are also not aware of any prior work that analyzes the privacy of learned Bloom filters.

2 Preliminaries

For a set S, $x \xleftarrow{\$} S$ denotes that $x \in S$ is sampled uniformly at random from S. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \cdots, n\}$. A Standard Bloom filter [Blo70] is a bit string $M = \{0,1\}^m$ of length m bits indexed over $[m]$, along with k different hash functions h_i. Each h_i maps an element from $x \in U$ to an index value within M, i.e, $h_i : U \mapsto [m]$. Let **SB** be a Standard Bloom filter. To encode a set S in **SB**, initialize a bit string M with all bits set to 0. Then, take each element $x \in S$, and for $i \in [k]$, set the bit corresponding to index $h_i(x)$ of M to 1. To answer a query for some element $x \in U$ in **SB**, return 1 if every bit in M corresponding to indices $h_i(x)$ is 1. Otherwise, return 0.

2.1 Naor-Oved-Yogev Model

The first major adversarial model for Bloom filters is developed in a series of papers by Naor et al. [NE19, NO22, LN25]. We will refer to this as the Naor-Oved-Yogev (NOY) Model in this paper. For a finite universe U of cardinality u, consider a set $S \subseteq U$. Naor and Yogev [NE19] define a Bloom filter as a data structure composed of two algorithms: a construction algorithm and a query algorithm.

Definition 1. Let $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2)$ be a pair of PPT algorithms. $\mathbf{B}_1$ takes a set $S \subseteq U$ and returns a representation M. $\mathbf{B}_2$ takes a representation M and a query element $x \in U$ and outputs a value in $\{0,1\}$. $\mathbf{B}$ is an (n, ε)-Bloom filter if for all sets $S \subseteq U$ of cardinality n, the following two properties hold.

1. Completeness: For any $x \in S$: $\Pr[\mathbf{B}_2(\mathbf{B}_1, x) = 1] = 1$
2. Soundness: For any $x \notin S$: $\Pr[\mathbf{B}_2(\mathbf{B}_1(S), x) = 1] \leq \varepsilon$

where the probabilities are taken over the random coins of $\mathbf{B}_1$ and $\mathbf{B}_2$ [NE19].

We assume $\mathbf{B}$ always has this format in this paper. If a Bloom filter's query algorithm cannot change the set representation, M, it is called a *steady* Bloom filter. Otherwise, it is called an *unsteady* Bloom filter. For simplicity, we assume a steady Bloom filter in our results (similar to prior work [NO22, LN25]), unless explicitly stated otherwise.

Naor and Oved [NO22] define $\texttt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$, a unified security game for Bloom filters. The game has an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. $\mathcal{A}_1$ chooses any set $S \subseteq U$. $\mathcal{A}_2$ takes set S and performs adaptive queries to a Bloom filter $\mathbf{B}$. $\mathcal{A}_2$ is also allowed oracle access to the query algorithm $\mathbf{B}_2$. λ is the security parameter, it is given to $\mathcal{A}_1$ and $\mathbf{B}_1$. t denotes the number of queries $\mathcal{A}_2$ is allowed to perform.

Definition 2. $\texttt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$ [NO22]

1. Adversary $\mathcal{A}_1$ takes $1^{\lambda + n \log u}$ and returns a set $S \subseteq U$ of cardinality n.
2. $\mathbf{B}_1$ takes $(1^{\lambda + n \log u}, S)$ and builds representation M.
3. Adversary $\mathcal{A}_2$ takes $(1^{\lambda + n \log u}, S)$ and oracle access to $\mathbf{B}_2(M, \cdot)$, and performs at most t adaptive queries $x_1, \cdots, x_t$ to $\mathbf{B}_2(M, \cdot)$.

There are many security notions based on AdaptiveGame. The two relevant to our work are the Always-Bet (AB) Test, proposed by Naor and Yogev [NE19], and a stronger notion called Bet-or-Pass (BP) Test, proposed by Naor and Oved [NO22].

2.1.1 Always-Bet (AB) Test

In the Always-Bet (AB) test, adversary $\mathcal{A}$ plays $\texttt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$ and is then required to return an $x^* \in U$. $\mathcal{A}$ wins if x^* is an unseen false positive.

AB Test $\texttt{ABTest}_{\mathcal{A},t}(\lambda)$ [NE19, NO22]:

1. $\mathcal{A}$ plays $\texttt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$. S is the set $\mathcal{A}$ chose in the game and $\{x_1, \cdots, x_t\}$ are the queries $\mathcal{A}$ performed in the game.
2. $\mathcal{A}$ returns $x^* \notin S \cup \{x_1, \cdots, x_t\}$.
3. If $\mathbf{B}_2(M, x^*) = 1$, return 1. Return 0, otherwise.

Definition 3. An (n, ε)-Bloom filter $\mathbf{B}$ is (n, t, ε)-AB test resilient if for any adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\Pr[\texttt{ABTest}_{\mathcal{A},t}(\lambda) = 1] \leq \varepsilon + \mathsf{negl}(\lambda)$$

where the probabilities are taken over the internal randomness of $\mathbf{B}$ and $\mathcal{A}$. [NE19, NO22]

Naor and Yogev [NE19] introduce a Bloom filter construction that is robust under the AB-test. Their construction is based on keyed pseudo-random permutations. Similar to other papers [ABT24], we will refer to this construction as the Naor-Yogev (NY) filter.

2.1.2 Bet-or-Pass (BP) Test

In the Bet-or-Pass (BP) test, adversary $\mathcal{A}$ can *pass* instead of returning an unseen false positive x^*. $\mathcal{A}$ plays $\texttt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$ and is then required to return (b, x^*). $b \in \{0, 1\}$ represents whether $\mathcal{A}$ wants to *bet* on the returned element x^*, or *pass*. $\mathcal{A}$'s win is based on a profit $C_{\mathcal{A}}$ defined by the test.

BP Test $\texttt{BPTest}_{\mathcal{A},t}(\lambda)$ [NO22]:

1. $\mathcal{A}$ plays $\texttt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$. S is the set $\mathcal{A}$ chose in the game and $\{x_1, \cdots, x_t\}$ are the queries $\mathcal{A}$ performed in the game.
2. $\mathcal{A}$ returns (b, x^*) where $x^* \notin S \cup \{x_1, \cdots, x_t\}$.
3. Return $\mathcal{A}$'s profit $C_{\mathcal{A}}$, defined as

$$C_{\mathcal{A}} = \begin{cases} \frac{1}{\varepsilon}, & \text{if } x^* \text{is a false positive and } b = 1, \\ -\frac{1}{1-\varepsilon}, & \text{if } x^* \text{is not a false positive and } b = 1, \\ 0, & \text{if } b = 0. \end{cases}$$

Definition 4. An (n, ε)-Bloom filter $\mathbf{B}$ is (n, t, ε)-BP test resilient if for any adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathbb{E}[C_{\mathcal{A}}] \leq \mathsf{negl}(\lambda)$$

where the probabilities are taken over the internal randomness of $\mathbf{B}$ and $\mathcal{A}$. [NO22]

The Bet-or-Pass test is the strongest security notion [LN25] currently defined in the $\texttt{AdaptiveGame}$ setting. Naor and Oved [NO22] prove that BP test is strictly stronger than AB test. Specifically, they prove that (n, t, ε)-BP test resilience implies (n, t, ε)-AB test resilience, and the converse implication is false.

Naor and Oved [NO22] introduce a Bloom filter construction that is robust under the BP-test. Their construction builds on an earlier Cuckoo hashing-based construction by Naor and Yogev [NE19] and relies on keyed pseudo-random functions. Similar to other papers [ABT24], we will refer to this construction as the Naor-Oved-Yogev (NOY) Cuckoo filter.

2.1.3 Universe and Adversary Types

A universe U is *small* if its cardinality $u \in \mathcal{O}(\mathsf{poly}(t, n, \lambda))$, otherwise U is *large*. An adversary with a query budget t can query at most a negligible fraction of the elements of a large universe. Adversary $\mathcal{A}$ in $\mathtt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$ can either be computationally bounded, i.e., running in probabilistic polynomial time (PPT), or computationally unbounded, i.e., not restricted to PPT but still bounded by the number of queries t. Consider (n, t, ε)-resilient Bloom filter $\mathbf{B}$ under the AB test or BP test. If $\mathbf{B}$ is (n, t, ε)-resilient for any polynomial number of queries $t \in \mathcal{O}(\mathsf{poly}(n, \lambda))$ under a computationally bounded adversary, $\mathbf{B}$ is called (n, ε)-*strongly-resilient* [NE19]. If $\mathbf{B}$ is resilient for at most t queries, under a computationally unbounded adversary, then $\mathbf{B}$ is called t-*resilient* [LN25].

2.2 Filić Model

The second major adversarial model for Bloom filters was introduced by Filić, Paterson, Unnikrishnan, and Virdia [FPUV22, VF24, FKKU25]. Our presentation of the model modifies Filić et al.'s original notation for easier comparison with the model of Naor et al. The Filić model allows inserting elements into a Bloom filter $\mathbf{B}$ after $\mathbf{B}$'s construction. We can define this in Naor's notation by adding a third polynomial time algorithm, $\mathbf{B}_3$, that does insertions.

Definition 5. Let $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)$ be a 3-tuple of PPT algorithms. $\mathbf{B}_1$ takes a set $S \subseteq U$ and returns a representation M. $\mathbf{B}_2$ takes a representation M and a query element $x \in U$ and outputs a value in $\{0, 1\}$. $\mathbf{B}_3$ takes a representation M and a query element $x \in U$ and outputs a new representation M' encoding the set $S \cup \{x\}$. $\mathbf{B}$ is an (n, ε)-insertable Bloom filter if for all sets $S \subseteq U$ of cardinality n and for at most ℓ insertions, the following four properties hold.

1. Completeness: For any $x \in S$: $\Pr[\mathbf{B}_2(\mathbf{B}_1, x) = 1] = 1$
2. Soundness: For any $x \notin S$: $\Pr[\mathbf{B}_2(\mathbf{B}_1(S), x) = 1] \leq \varepsilon$
3. Element Permanence [FPUV22]: For any $x \in U$ and any M such that $\mathbf{B}_2(M, x) = 1$, if M' is a later state after any sequence of insertions, it must hold that $\mathbf{B}_2(M', x) = 1$.
4. Non-decreasing membership probability [FPUV22]: For any $x \in U$ and any M, let $M' = \mathbf{B}_3(M, x)$. For all $y \in U$, it must hold that $\Pr[\mathbf{B}_2(M', y)] \geq \Pr[\mathbf{B}_2(M, y)]$.

where the probabilities are taken over the random coins of $\mathbf{B}_1, \mathbf{B}_2$, and $\mathbf{B}_3$.

Filić's model uses a simulation-based definition for adversarial correctness. Their adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has two components similar to Naor's model. $\mathcal{A}_1$ chooses any set $S \subseteq U$. $\mathcal{A}_2$ takes set S and performs both adaptive queries and adaptive insertions to a Standard Bloom filter $\mathbf{SB}$. $\mathcal{A}_2$ is allowed oracle access to $\mathbf{B}_2(M, \cdot)$ and $\mathbf{B}_3(M, \cdot)$. $\mathcal{A}_2$ is also allowed access to an oracle O_M that returns the internal representation M of $\mathbf{SB}$. We first discuss Filić et al.'s ideal simulator and then discuss their adversarial correctness notion.

2.2.1 Ideal Simulator

Filić show that Standard Bloom filter constructions have two properties, function decomposability and reinsertion invariance, that can be used to reason about their performance in an honest setting without having to specify a particular input distribution[1]. Let $\mathbf{SB}$ be

[1]While the simulator's behavior is described below, these two properties provide the theoretical foundation for why such non-adversarial simulation is possible. See Section 3 of [FPUV22] for a detailed treatment.

$\texttt{Real}(\mathcal{A}, \mathsf{Sim}, \mathsf{D})$
1: Adversary $\mathcal{A}_1$ returns a set $S \subseteq U$.
2: $\mathbf{B}_1$ takes S and builds representation M.
3: $out \leftarrow\!\!\$\ \mathcal{A}_2^{\mathsf{O}}(S)$ where $\mathsf{O} = \{\mathbf{B}_2(M, \cdot), \mathbf{B}_3(M, \cdot), \mathsf{O}_M\}$
4: $d \leftarrow\!\!\$\ \mathsf{D}(out)$
5: **return** d.

$\texttt{Ideal}(\mathcal{A}, \mathsf{Sim})$
1: $out \leftarrow\!\!\$\ \mathsf{Sim}(\mathcal{A})$
2: $d \leftarrow\!\!\$\ \mathsf{D}(out)$
3: **return** d.

Figure 1: Real and Ideal experiments in the Filić model.

a Standard Bloom filter construction initialized with an empty set. Its behavior under an honest setting can be modelled using an algorithm n-NAI-gen that uniformly randomly samples n unique elements from U, inserts them into **SB**, and returns the final representation M of **SB** after all n insertions.

In the ideal world, $\mathcal{A}$ interacts with a simulator Sim which provides a non-adversarially-influenced view of **SB**'s behavior. Sim maintains its own internal state which contains

1. A representation M, which is bit string of length m just like a Standard Bloom filter.
2. A truly random function f which maps any element $x \in U$ to k indices in $[m]$.
3. Lists `inserted` and `FPList`, for elements confirmed to be inserted and elements identified as false positives respectively.
4. An integer counter, `ctr`, that keeps track of distinct insertions.

Sim implements oracles for $\mathbf{B}_1, \mathbf{B}_2(M, \cdot), \mathbf{B}_3(M, \cdot)$, and O_M in the following way. When given a set $S \subseteq U$, $\mathbf{B}_1$ computes k indices $f(x)$ for each $x \in S$, and sets the bit corresponding to each index in M to 1. It also adds x to the `inserted` list and increments `ctr` for each x. When given an element $x \in U$ as input, $\mathbf{B}_2$ returns 1 if x is in `inserted` or `FPList`. Otherwise, it samples k indices uniformly randomly from $[m]$ (it disregards x). If all k indices in M are set to 1, $\mathbf{B}_2$ adds x to `FPList` and returns 1. Otherwise, it returns 0. When given an element $x \in U$ as input, $\mathbf{B}_3$ does nothing if x is in `inserted`. Otherwise, $\mathbf{B}_3$ updates M by setting all the bits corresponding to the k indices returned by $f(x)$ to 1. It then adds x to `inserted` and increments `ctr`. O_M simply returns M.

Since Sim queries on random indices instead of the given element x, Sim's response only reflects the underlying density of the bit string M. This is precisely the false positive probability under an honest setting.

2.2.2 Adversarial Correctness Notion

In the Filić adversarial model, a security experiment a bit b is flipped and based on the output, the adversary $\mathcal{A}$ plays in either the real world ($b = 0$) or the ideal world ($b = 1$). After its interactions with either world are complete, $\mathcal{A}$ must return an output *out* that is given to a distinguisher D. The experiment then returns D's output. In the ideal world, $\mathcal{A}$ interacts with the ideal simulator defined above. In the real world, it is given oracle access to the algorithms of a real Standard Bloom filter construction **SB**. See Figure 1 for the real and ideal experiments.

Filić et al.'s adversarial correctness notion is a bound on the distinguisher D's probability of distinguishing between the real and the ideal world. To clearly distinguish this adversarial correctness notion from Naor et al.'s adversarial correctness notions, we will refer to it as Filić correctness.

Definition 6. Let $\mathbf{B}$ be an insertable Bloom filter. $\mathbf{B}$ is $(q_u, q_t, q_v, t_a, t_s, t_d, \varepsilon)$-Filić-correct if for all adversaries $\mathcal{A}$ running in time at most t_a, and making at most q_u, q_t, q_v queries to the oracle for $\mathbf{B}_3(M, \cdot)$, the oracle for $\mathbf{B}_2(M, \cdot)$, and O_M respectively with an ideal simulator Sim that runs in time at most t_s, and for all distinguishers D running in time at most t_d, we have

$$|\Pr[\mathtt{Real}(\mathcal{A}, \mathsf{Sim}, \mathsf{D}) = 1] - \Pr[\mathtt{Ideal}(\mathcal{A}, \mathsf{Sim}, \mathsf{D}) = 1]| \leq \varepsilon$$

2.3 Learned Bloom filters

Almashaqbeh, Bishop, and Tirmazi [ABT24] extend the NOY model to create an adversarial model for learned Bloom filters. We will refer to this as the Almashaqbeh-Bishop-Tirmazi (ABT) model in this paper. A learned Bloom filter is a Bloom filter that is working in collaboration with a learning model acting as a pre-filter. In this context, a regular Bloom filter, i.e., one that is not *learned* is referred to as a *classical* Bloom filter. Learned Bloom filters reduce the false positive rate of a Classical Bloom filter while maintaining the guarantee of no false negatives. A learned Bloom filter $\mathbf{LB}$ trains its learning model over the dataset $\mathbf{LB}$ represents, such that the model determines a function $\mathcal{L}$ that models this set. On input $x \in U$, $\mathcal{L}$ outputs the probability that $x \in S$, where S is the input set. Relevant definitions from the ABT model are stated below.

Definition 7. Let $S \subseteq U$ be any set encoded by a Bloom filter. For any two sets $P \subseteq S$ and $N \subseteq U \setminus S$, the training dataset is the set $\mathcal{T} = \{(x_i, y_i = 1) \mid x_i \in P\} \cup \{(x_i, y_i = 0) \mid x_i \in N\}$. [ABT24]

Definition 8. For an $\mathcal{L} : U \mapsto [0, 1]$ and threshold τ, we say $\mathcal{L}$ is an $(S, \tau, \varepsilon_p, \varepsilon_n)$-learning model, if for any set $S \subseteq U$ the following two properties hold:

1. P-Soundness: $\forall x \notin S : \Pr[\mathcal{L}(x) \geq \tau] \leq \varepsilon_p$
2. N-Soundness: $\forall x \in S : \Pr[\mathcal{L}(x) < \tau] \leq \varepsilon_n$

where the probability is taken over the random coins of $\mathcal{L}$. [ABT24]

In the ABT model, a learned Bloom filter is defined in the following way. Similar to Naor and Oved [NO22], Almashaqbeh et al. only consider steady Bloom filters in which the query algorithm $\mathbf{B}_2$ does not change either the classical representation M or the learned representation $(\mathcal{L}, \tau)$ of the input set S.

Definition 9. A learned Bloom filter $\mathbf{LB} = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4)$ is a 4-tuple of PPT algorithms: $\mathbf{B}_1$ is a construction algorithm, $\mathbf{B}_2$ is a query algorithm, $\mathbf{B}_3$ is a randomized algorithm that takes a set $S \subseteq U$ as input and outputs a training dataset $\mathcal{T}$, and $\mathbf{B}_4$ is a randomized algorithm that takes the training dataset $\mathcal{T}$ as input and returns a learning model $\mathcal{L}$ and a threshold $\tau \in [0, 1]$. The internal representation of $\mathbf{LB}$ contains two components: the classical component M and the learned component $(\mathcal{L}, \tau)$. $\mathbf{B}_2$ takes as inputs an element $x \in U$, M, and $(\mathcal{L}, \tau)$, and outputs 1 indicating that $x \in S$ and 0 otherwise. We say that $\mathbf{B}$ is an $(n, \tau, \varepsilon, \varepsilon_p, \varepsilon_n)$-learned Bloom filter if for all sets $S \subseteq U$ of cardinality n, it holds that

1. Completeness: $\forall x \in S : \Pr[\mathbf{B}_2(\mathbf{B}_1(S), \mathbf{B}_4(S, \mathbf{B}_3(S)), x) = 1] = 1$.
2. Filter soundness: $\forall x \notin S : \Pr[\mathbf{B}_2(\mathbf{B}_1(S), \mathbf{B}_4(S, \mathbf{B}_3(S)), x) = 1] \leq \varepsilon$.
3. Learning model soundness: $\mathbf{B}_4(S, \mathbf{B}_3(S))$ is an $(S, \tau, \varepsilon_p, \varepsilon_n)$-learning model.

where the probabilities are over the random coins of $\mathbf{B}_1$, $\mathbf{B}_3$, and $\mathbf{B}_4$. [ABT24]

Almashaqbeh et al. extend Naor et al.'s AB-test and BP-test to create versions suitable for learned Bloom filters. They refer to the learned versions of these tests as Learned-Always-Bet (LAB) and Learned-Bet-or-Pass (LBP), respectively. Almashaqbeh et al. introduce two learned Bloom filter constructions that are robust under LAB and LBP, respectively. Their constructions combine Naor et al.'s classical Bloom filter constructions with partitioned learned Bloom filters [VKMK21]. We will refer to these constructions as the ABT filter and the ABT Cuckoo filter.

2.4 Randomized Response

Warner's randomized response is one of the most common private set membership techniques, first proposed in 1965 [War65]. Scientists have used it to survey set membership among a population for things that individual members wish to retain confidentiality about. A commonly used example is "Are you a member of the Communist Party?" [HLM08]. Other examples include surveying the number of abortion recipients [AGH70] and surveys regarding sexual orientation [XQZ+14]. In Warner's randomized response, the respondent answers a Yes/No question truthfully with probability p. With probability $1 - p$, the respondent flips a fair coin. The respondent answers *Yes* if the coin is heads, and *No* otherwise. Thanks to this technique each respondent has *plausible deniability* regarding their membership. Mangat [Man94] proposed a variant of Warner's randomized response. In Mangat's randomized response, a respondent answers truthfully to a Yes/No question with probability p. With probability $1 - p$, the respondent always answers Yes. Unlike Warner's randomized response, Mangat's variant only introduces one-sided error into the dataset.

3 Open Problems

Naor et al. [NO22, LN25] introduce a hierarchy of game-based security notions for Bloom filters, as we discussed in Section 2. Separately, Filić et al. [FPUV22, FKKU25] introduce an alternate simulator-based security definition. Filić et al.'s security notion guarantees that the false positive rate observed by an adversary has a low probability of being significantly larger than the false positive rate observed in a non-adaptive setting.

Both Naor et al. and Filić et al.'s security notions share the same intuitive goal, i.e, minimizing false positives. However, no formal connection is currently known between them. Naor and Lotan [LN25] leave this as an open direction in their paper. Understanding the connection between these two approaches would help unify the robustness literature and clarify which notions provide stronger guarantees in practice.

Problem 1. There are two dominant security frameworks for Bloom filters: Naor et al.'s game-based notions and Filic et al.'s simulator-based notion. Are there provable connections between the two frameworks?

Another key distinction between the Naor and Filić models is that the NOY model does not allow the adversary to insert elements into a Bloom filter after construction, while the Filić model does allow insertions. Is it possible to create *dynamic* versions of Naor et al.'s security notions, i.e., the AB-test and BP-test, etc., that allow insertions? For example, enabling an adversary to interleave insertions and queries before betting?

Problem 2. Can Naor et al.'s security games be generalized to insertable Bloom filters?

Almashaqbeh et al. [ABT24] extend the NOY model to support learned Bloom filters. They also propose learned Bloom filter constructions that are robust under learned versions of the AB-test and BP-test, respectively. However, it is unknown whether Almashaqbeh et al.'s robust learned Bloom filter constructions also satisfy Filić correctness. It is still

undetermined whether Filić et al.'s adversarial model is compatible with learned Bloom filters and if secure constructions exist that satisfy the Filić correctness notion.

Problem 3. Can the Filić adversarial model be extended to learned Bloom filters? Do learned Bloom filter constructions exist that satisfy the notion of Filić correctness?

Similar to the classical Bloom filter constructions of Naor et al. [NE19, NO22, LN25], the learned Bloom filter constructions of Almashaqbeh et al. do not allow an adversary to insert elements into the Bloom filter after construction. Therefore, just like for Naor et al.'s classical Bloom filter constructions, further work is required to understand whether Almashaqbeh et al.'s learned Bloom filter constructions support any robustness notions for insertable Bloom filters. In fact, whether or not there exists a robust (under any notion) learned Bloom filter that allows inserts after construction is itself an unsolved problem.

Problem 4. Are there any insertable learned Bloom filter construction that provably satisfies a meaningful robustness notion?

Naor and Oved [NO22] raise a question regarding the BP-test resilience of a Standard Bloom filter that uses keyed pseudo-random functions F_i instead of public hash functions h_i. They write "Note that it is not known whether replacing the hash functions with a PRF in the standard construction of Bloom filters (i.e., the one in the style of Bloom's original one [Blo70]) results in a Bloom filter that is BP test resilient." [NO22]. This is an important question because the Standard Bloom filter construction is still one of the most widely deployed Bloom filter constructions. For example, it is deployed in the Linux Kernel [Fou23] and Google's LevelDB [Goo23]. We precisely define the problem below.

Problem 5. Let **MB** be a modified construction of a Standard Bloom filter **SB** that replaces each hash function h_i in **SB** with a keyed PRF F_i. Does **MB** satisfy (n, ε)-strong resilience under the BP-test?

As we discussed in the introduction, many works [SBBR17, RSMW+22, GRW+22] have shown that Bloom filters leak information regarding the stored input set. An open problem is exploring rigorous privacy guarantees based on the notion of differential privacy for Bloom filters.

Problem 6. Are there any Bloom filter constructions that provide rigorous differential privacy guarantees for the set they store?

The robustness tests under the NOY model, including the AB-test and the BP-test, assume that the adversary makes *distinct* queries. Lotan and Naor [LN25] (and Naor and Oved [NO22] but with less details) pose an open problem regarding the robustness of Bloom filters when query repetition is allowed. Lotan and Naor's proposed direction can be broken down into three precise questions.

Problem 7. Does there exist a Bloom filter construction that satisfies BP-test resilience when the adversary is allowed to repeat queries?

Note that this requires extending the BP-test definition to handle repeated queries instead of only allowing distinct queries. Bender et al. [BFCG+18] introduce a construction called a broom filter that has provable guarantees under repeated queries. In their paper, Bender et al. introduce their adversarial model for Bloom filters, which we will refer to as the Bender model. Unlike the NOY model, which only allows distinct queries, the Bender model allows repeated queries. Lotan and Naor also ask whether the NOY model is compatible with the Bender model or has provable connections.

Problem 8. Are there any provable connections between Naor et al.'s security notions, which do not allow query repetition, and Bender et al.'s adversarial model, which does allow query repetition?

Mitzenmacher et al. [MPR20] provide a Bloom filter construction called an Adaptive Cuckoo Filter that removes false positives after they are queried. However, it is not known whether Adaptive Cuckoo Filters are provably adaptive [BFCG+18] under any of the discussed adversarial models.

Problem 9. Are there any provable bounds on the adversarial robustness of Adaptive Cuckoo Filters under a known Bloom filter adversarial model?

Finally, there is another well-known adversarial model for Bloom filters, introduced by Clayton, Patton, and Shrimpton [CPS19]. The Clayton-Patton-Shrimpton (CPS) model extends the NOY model using a similar game-based formalism. Recall that Naor et al.'s AB-test allows an adversary to make t distinct adaptive queries, before outputting a new, unseen challenge query. Only this challenge query needs to be a false positive for the adversary to win. Clayton et al. instead allow an adversary to win if the adversary can forge a number of distinct false positive queries during its entire execution that is above a parametrized threshold. A formal reduction or separation between the CPS and NOY models would clarify their comparative strengths and applicability across Bloom filters.

Problem 10. Are there any provable connections between the NOY model and the CPS model?

3.1 Taxonomy

We present a taxonomy of the open problems we discussed that unifies the contributions of recent work on the adversarial robustness of Bloom filters. Our classification contains three axes: robustness notions, construction features, and model relationships. We provide a table for each axis, with open problems indicated with $\diamond$ in the tables.

Robustness Notions: provable guarantees for Bloom filters differ significantly across definitions, with two dominant families of adversarial models. The first family encompasses game-based notions, including those that cover learning-based robustness. Naor et al. [NE19, NO22, LN25], Clayton et al. [CPS19], and Almashaqbeh et al. [ABT24] define adversarial correctness in terms of win conditions in an interactive game. The second family relies on simulator-based notions. The only current notable example of this is the work of Filić et al. [FPUV22, FKKU25]. Privacy-based notions for Bloom filters remain less well-explored. Filić et al. [FPUV22] propose a simulator-based privacy definition for Bloom filters based on information leakage. We summarize prior work in terms of this axis in Table 1.

Construction features: we map known secure Bloom filter constructions to the robustness notions they satisfy in Table 2. This includes constructions that are learned or classical, and insertable or static (i.e., no post-construction updates). The majority of open problems on this axis relate to extending robustness guarantees to learned, insertable, or repeated query settings.

Model relationships: Table 3 summarizes the space of known and unknown connections between adversarial models. Note that this does not include connections between notions within the same model, such as those explored in the work of Naor and Oved [NO22] or Almashaqbeh et al. [ABT24]. The vision here is for the community to incrementally develop a single unified adversarial model that captures all security notions.

4 Private Bloom filters

In this section, we attempt to solve Problem 6 by providing two constructions for Bloom filters with differential privacy guarantees. We first discuss how to adapt differential privacy [DR+14] and asymmetric differential privacy [TKCY22] for unordered sets and

Table 1: Mapping of robustness notions to Bloom filter classes

Notion	Classical Bloom Filter	Learned Bloom Filter	Insertable Bloom Filter	Repeated Queries
NOY AB	[NO22]	◇	◇	◇
NOY BP	[NO22]	◇	◇	◇
Filić	[FPUV22]	◇	[FPUV22]	◇
ABT LAB	◇	[ABT24]	◇	◇
ABT LBP	◇	[ABT24]	◇	◇
Bender	[BFCG+18]	◇	[BFCG+18]	[BFCG+18]
Diff. Privacy	✓	✓	◇	✓

Legend: ✓ = Contribution of this paper; ◇ = Open problem.

Table 2: Robustness and privacy notions satisfied by each relevant Bloom filter construction.

Construction	Learned	Insert.	Naor AB	Naor BP	Filić	Rep. Queries	Diff. Priv.
SBF	N	Y	N	N	N	N	N
PRF-Backed SBF	N	N	◇	×	◇	◇	◇
NY	N	N	Y	N	◇	◇	◇
NOY Cuckoo	N	N	Y	Y	◇	◇	◇
FPUV	N	N	✓	◇	Y	◇	◇
ABT	Y	N	Y	N	◇	◇	◇
ABT Cuckoo	Y	N	Y	Y	◇	◇	◇
Broom	N	Y	◇	◇	◇	Y	◇
Mangat	✓	N	◇	◇	◇	◇	✓
Warner	✓	N	◇	◇	◇	◇	✓

Legend: Y/N = yes / no result by prior work; ✓ / × = yes / no result contributed by this paper; ◇ = open problem.

Table 3: Summary of known connections between adversarial models

Model	NOY	Filić	ABT	Bender	CPS
NOY	∗	✓	◇	◇	◇
Filić	✓	∗	◇	◇	◇
ABT	◇	◇	∗	◇	◇
Bender	◇	◇	◇	∗	◇
CPS	◇	◇	◇	◇	∗

Legend: ✓ = provable connections contributed by this paper; ◇ = open problem; ∗ = trivially true.

show that Mangat's randomized response (Section 2.4) satisfies asymmetric differential privacy. We discuss two Private Bloom filter constructions, the Mangat filter and the Warner filter. We then investigate the error rates of the private Bloom filter constructions as compared to the Standard Bloom filter construction. Finally, we discuss how the private Bloom filter constructions are also applicable to learned Bloom filters.

4.1 Differential Privacy on Unordered Sets

For any two sets A, B, we can use an unweighted version of the Jaccard distance, $d_{sj}(A, B) = |A \cup B| - |A \cap B|$, to measure set similarity. The well-known notions of symmetric [DR+14] and asymmetric [TKCY22] differential privacy can then be written in

terms of unordered sets in the following way.

Definition 10. A randomized algorithm $\mathbf{A}_r$ satisfies (ϵ, δ)-differential privacy [DR+14] if for any two sets S, S' s.t $d_{sj}(S, S') \leq 1$ and for any possible output range $O \subseteq \text{Range}(\mathbf{A}_r)$,

$$P[\mathbf{A}_r(S) \in O] \leq e^{\epsilon} P[\mathbf{A}_r(S' \in O] + \delta$$

where the probabilities are over $\mathbf{A}_r$.

Definition 11. A randomized algorithm $\mathbf{A}_r$ satisfies $(\varepsilon, \varepsilon', \delta)$-asymmetric differential privacy [TKCY22] if for any two sets S, S' such that $d_{sj}(S, S') \leq 1$, and for any possible output range $O \subseteq \text{Range}(\mathbf{A}_r)$, the following two properties hold:

1. If $S' = S \setminus \{x\}$ for some $x \in S$, then $\Pr[\mathbf{A}_r(S) \in O] \leq e^{\varepsilon} \Pr[\mathbf{A}_r(S') \in O] + \delta$
2. If $S' = S \cup \{x\}$ for some $x \notin S$, then $\Pr[\mathbf{A}_r(S) \in O] \leq e^{\varepsilon'} \Pr[\mathbf{A}_r(S') \in O] + \delta$

where probabilities are over $\mathbf{A}_r$.

Asymmetric differential privacy aligns well with many known set membership scenarios where presence in a set is sensitive, while absence is not. For example, knowing that an individual belongs to the set of Communist party members, HIV patients, or recipients of abortions can be highly sensitive, whereas knowing that an individual is not in these sets often does not reveal sensitive information. There are also situations where this type of privacy guarantee is *necessary*. For example, in epidemic analysis, when creating a set of the number of infected individuals that visited a location [TKCY22], having a two-sided error is not useful.

There is a well-known result [DR+14] that demonstrates the differential privacy of Warner's randomized response. When the probability of the respondent answering a question truthfully is p, Warner's randomized response satisfies $(\ln\left(\frac{p}{1-p}\right), 0)$-differential privacy. This result also holds for differential privacy when applied to sets. We now show that Mangat's randomized response satisfies asymmetric differential privacy for sets. This will be needed for constructing private Bloom filters.

Theorem 1. Mangat's randomized response satisfies $(\ln(\frac{1}{1-p}), \ln(1-p), 0)$-asymmetric differential privacy.

Proof. Let S, S' be two sets s.t $d_{sj}(S, S') = 1$, and $\mathbf{A}_r$ be Mangat's randomized response. The probabilities are taken over $\mathbf{A}_r$. First, take the case where $S' \subset S$, i.e., $S' = S \setminus \{x\}$ for some $x \in S$.

$$\frac{\Pr[\mathbf{A}_r(S) \in O]}{\Pr[\mathbf{A}_r(S') \in O]} = \frac{\Pr[x \in \mathbf{A}_r(S)]}{\Pr[x \in \mathbf{A}_r(S')]} = \frac{1}{1-p}$$

Hence, $\varepsilon = \ln(\frac{1}{1-p})$. Now take the case where $S \subset S'$, i.e., $S' = S \cup \{x\}$ for some $x \notin S$.

$$\frac{\Pr[\mathbf{A}_r(S) \in O]}{\Pr[\mathbf{A}_r(S') \in O]} = \frac{\Pr[x \in \mathbf{A}_r(S)]}{\Pr[x \in \mathbf{A}_r(S')]} = 1 - p$$

Therefore $\varepsilon' = \ln(1-p)$ and $\delta = 0$. The result follows. □

4.2 Mangat and Warner filters

Since Bloom filters execute randomized algorithms to store sets, the set privacy notions can be modified to get analogous Bloom filter privacy notions.

Definition 12. An (n, ε)-Bloom filter $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2)$ is an $(n, \varepsilon, \varepsilon_p, \delta_p)$-private Bloom filter if for all $S, S' \subseteq U$ such that $d_{sj}(S, S') \leq 1$ and for all representations M, $\Pr[\mathbf{B}_1(S) = M] \leq e^{\varepsilon_p} P[\mathbf{B}_1(\S') = M] + \delta_p$ where the probabilities are over the coins of C_r.

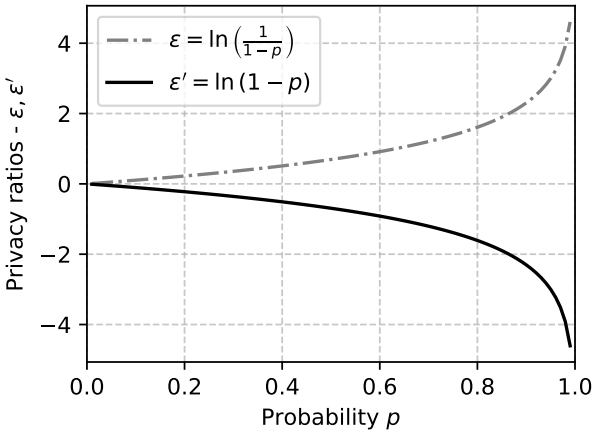

Figure 2: The asymmetric privacy bounds of a Mangat filter as p varies. ε is the privacy of an element not in the output set, while ε' is the privacy of an element in the output set.

An $(n, \varepsilon, \varepsilon_p, \varepsilon'_p, \delta_p)$-asymmetric private Bloom filter can be defined analogously using the asymmetric differential privacy definition for sets.

We now introduce two private Bloom filter constructions, Mangat filters and Warner filters, based on Mangat and Warner randomized response respectively. Mangat filters keep a Bloom filter's traditional one-sided guarantees, i.e., no false negatives only false positives. However, Mangat filters only satisfy asymmetric differential privacy. Warner filters satisfy (symmetric) differential privacy, at the cost of returning false negatives with a small probability.

A Mangat filter **MB** can be constructed from any Bloom filter $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2)$ in the following way. Replace $\mathbf{B}_1$ with $\mathbf{B}'_1$ that works in the following way. Fix a probability $p \in (0, 1)$. Initialize $S' \leftarrow S$. For each $x \in U \setminus S$, add x to S' with probability $1 - p$. Call the original construction algorithm $\mathbf{B}_1$ on S' instead of S, i.e., return $\mathbf{B}_1(S')$. The query algorithm $\mathbf{B}_2$ remains unchanged.

Theorem 2. Mangat filter satisfies $(\ln(\frac{1}{1-p}), \ln(1-p), 0)$-asymmetric differential privacy.

Proof. A Mangat filter is a special case of Mangat's randomized response mechanism applied to set membership. Let S and S' be two sets s.t $d_{sj}(S, S') \leq 1$. A Mangat filter modifies input set S by adding each element $x \in U \setminus S$ with probability p. This follows the structure of Mangat's randomized response mechanism from Theorem 1 and satisfies the given same asymmetric differential privacy guarantee. □

Theorem 2 proves that a Mangat filter satisfies *asymmetric* differential privacy by adding elements with a controlled probability $1-p$. Asymmetric differential privacy enables us to explicitly model scenarios where an adversary $\mathcal{A}$'s ability to infer the presence of an element in the original set is significantly weaker than $\mathcal{A}$'s ability to infer the absence of an element in the original set. We illustrate this in Figure 2. ε and ε' model $\mathcal{A}$'s ability to infer the absence and presence of an element in the original set, respectively. Since a Mangat filter never removes an element in the original set, ε does not meaningfully constrain A's ability to infer absence. ε', however, provides meaningful privacy for presence. The privacy increases as $\varepsilon' \to 0$ which happens as $1 - p \to 1$, i.e., $p \to 0$. Intuitively, when all elements in the universe appear in the output set, $\mathcal{A}$ has little probability of distinguishing which elements were in the original set. The privacy decreases as $\varepsilon' \to -\infty$ $(1 - p \to 0)$, i.e, as the Mangat filter probabilistically adds fewer elements.

A Warner filter **WB** can be constructed for a Bloom filter by modifying its construction algorithm by replacing $\mathbf{B}_1$ with $\mathbf{B}'_1$ that works as follows. Fix a probability $p \in (\frac{1}{2}, 1)$. Initialize $S' \leftarrow \emptyset$. For each $x \in S$, add x to S' with probability p. For each $x \in U \setminus S$, add x to S' with probability $1 - p$.

Theorem 3. Warner filter satisfies $\left(\ln\left(\frac{p}{1-p}\right), 0\right)$-differential privacy.

Proof. A Warner filter is a special case of Warner's randomized response mechanism applied to set membership. Let S and S' be two sets s.t $d_{sj}(S, S') \leq 1$. A Warner filter modifies input set S by removing each element $x \in S$ with probability p and adding each element $x \in U \setminus S$ with probability $1 - p$. This follows the structure of Warner's randomized response mechanism and therefore we can directly apply the well-known result [DR+14] that Warner's randomized response satisfies the given differential privacy guarantee. □

4.3 Error Rate Analysis

We now investigate how our method for adding privacy affects the false positive rate (FPR) and its false negative rate (FNR) of a given Standard Bloom filter **SB**. We do not classify queries to elements added by our privacy-preserving algorithms as false positives, i.e., a query on $x \in S' \setminus S$ is not a false positive. These elements are not representative of typical false positives, which arise naturally due to the probabilistic nature of the **SB**. As such, we exclude these elements from the FPR calculations to focus on the inherent accuracy of the **SB** under privacy-preserving conditions. This distinction ensures a clear separation between errors resulting from **SB**'s one-sided guarantees and those intentionally introduced for privacy purposes.

Assume **SB** stores set $S \subseteq U$, has internal bit-string M, and k hash functions. Let $\text{FPR}(S, M, k)$ and $\text{FNR}(S, M, k)$ be functions that return the expected FPR and expected FNR of **SB**, respectively. Then the FPR and FNR of a private Bloom filter built on top of **SB** that constructs set S' from the original set S will be $\text{FPR}(S', M, k)$ and $\text{FNR}(S', M, k)$ respectively. It is well-known that **SB** has approximately the following false positive rate [BM04], $\text{FPR}(S, M, k) = (1 - e^{-k \cdot |S|/|M|})^k$ where $|M|$ is the length of the bit-string M. For a given set S, the expected cardinality of the set S' stored by a Mangat filter is $|S'| = |S| + (1 - p)(|U| - |S|)$. Similarly, for a Warner filter, it is $|S'| = |S| + p(|U| - |S|) - (1 - p)|S|$. We can replace $|S|$ in the FPR equation for **SB** with these expressions to get FPR expressions for private Bloom filters. When using the Warner filter, we will also have a non-zero FNR, which is simply the probability that a given $x \in S$ is not included in the set S' by the construction algorithm, i.e, $1 - p$.

4.4 Applicability to Learned Bloom filters

Mangat and Warner filters modify the input set S prior to Bloom filter construction, without altering the structure of the Bloom filter itself. As a result, both constructions are fully compatible with learned Bloom filters, including those modeled in the ABT framework [ABT24]. A learned Bloom filter $\mathbf{LB} = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4)$ relies on a training dataset $\tau = \mathbf{B}_3(S)$ generated from the input set S. Since the Mangat and Warner filters perturb S to produce a new set S', the learning model is trained on $\tau' = \mathbf{B}_3(S')$ instead of τ. Mangat and Warner filters thus act as a privacy-preserving pre-processing step on the input set before training and construction.

In other words, if **LB**'s underlying learning model and classical Bloom filter(s) satisfy standard correctness guarantees over S', and if S' is generated using a randomized response mechanism satisfying symmetric or asymmetric differential privacy, then the overall construction **LB** inherits the same privacy guarantees. No modification to the data structure and its underlying algorithms is required. This observation allows Warner and

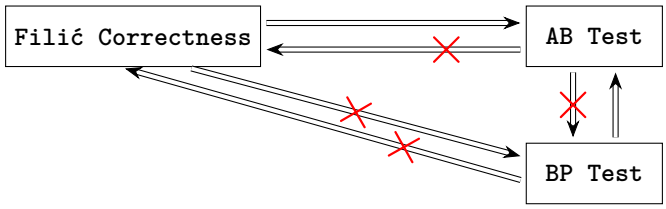

Figure 3: Connections between Naor and Filić notions.

Mangat filters to serve as generic wrappers for private learned Bloom filters, expanding the scope of private Bloom filter constructions beyond classical Bloom filters. To our knowledge, this is the first approach that provides provable differential privacy guarantees for learned Bloom filters.

5 Bridging NOY and Filić Models

In this section, we solve Problem 1 by providing provable connections between the robustness notions of Naor et al.'s model and Filić et al.'s model. Lotan and Naor [LN25] provide a counter-example demonstrating that Filić correctness does not imply resilience under the BP test. Our work shows that Filić correctness *does* imply resilience under the AB test. We also use a counter-example construction introduced by Almashaqbeh et al. [ABT24] to demonstrate that resilience under AB test or BP test does not imply Filić correctness.

Theorem 4. If a Bloom filter $\mathbf{B}$ is $(q_u, q_t, q_v, t_a, t_s, t_d, \varepsilon)$-Filić-correct then $\mathbf{B}$ is also $(n, q_t - 1, 2\varepsilon)$-resilient under the AB test for adversaries running in time at most t_a.

Proof. Assume $\mathbf{B}$ is not $(n, q_t - 1, 2\varepsilon)$-resilient under the AB test, i.e., there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ running in time at most t_a who can win the AB test with probability non-negligibly greater than ε. We will show how to construct an adversary $\mathcal{A}'$ using $\mathcal{A}$ and a distinguisher D that distinguishes between the Real and Ideal worlds in the Filić experiment with probability greater than ε.

$\mathcal{A}'$ plays the experiments in the Filić model in the following way. $\mathcal{A}'$ runs $\mathcal{A}_1$ to get set S which it forwards to the Filić experiment. $\mathcal{A}_2$ requires oracle access to $\mathbf{B}_2(M, \cdot)$ to make $q_t - 1$ adaptive queries. $\mathcal{A}'$ forwards $\mathcal{A}_2$'s queries to the $\mathbf{B}_2(M, \cdot)$ oracle provided to $\mathcal{A}'$ in the Filić experiment. After $q_t - 1$ adaptive queries, $\mathcal{A}_2$ returns x^* as required by the AB test. $\mathcal{A}'$ uses the last query in its q_t query budget for oracle $\mathbf{B}_2(M, \cdot)$ to get $out = \mathbf{B}_2(M, x^*)$ and returns out as its output. Distinguisher D outputs $d = out$, i.e., it decides it is in the real world if $out = 1$ and in the ideal world otherwise.

Since we assumed $\mathbf{B}$ is not $(n, q_t - 1, 2\varepsilon)$-resilient under the AB test, $\Pr[\mathtt{Real}(\mathcal{A}, \mathsf{Sim}, \mathsf{D}) = 1] > 2\varepsilon + \mathsf{negl}(\lambda)$. Since the ideal simulator, Sim, ignores $\mathcal{A}$'s output x^* and picks k indices uniformly randomly, $\mathcal{A}$'s choice has no impact on Sim's false positive probability. Sim's false positive probability is $\mathbf{B}$ false positive probability in a non-adversarial setting, which is at most $\mathbf{B}$'s false positive probability in an adversarial setting, i.e, $\Pr[\mathtt{Ideal}(\mathcal{A}, \mathsf{Sim}, \mathsf{D}) = 1] \leq \varepsilon$. Therefore $|\Pr[\mathtt{Real}(\mathcal{A}, \mathsf{Sim}, \mathsf{D}) = 1] - \Pr[\mathtt{Ideal}(\mathcal{A}, \mathsf{Sim}, \mathsf{D}) = 1]| > |2\varepsilon + \mathsf{negl}(\lambda) - \varepsilon| > \varepsilon$, and therefore the distinguishing advantage is larger than ε violating Filić correctness. Hence, we have shown that $\mathbf{B}$ is not $(n, q_t - 1, 2\varepsilon)$-resilient under the AB test it is also not $(q_u, q_t, q_v, t_a, t_s, t_d, \varepsilon)$-Filić-correct. The result follows. □

The converse does not hold, i.e., resilience under the AB test does not imply Filić correctness. A trivial counter-example is a NY filter, which is resilient under the AB test [NO22]. A NY filter uses a Standard Bloom filter and a keyed pseudo-random permutation $\mathsf{PRP}_{\mathsf{sk}}$ with secret key sk. For any element $x \in U$, the NY filter stores and

queries $\mathsf{PRP}_{\mathsf{sk}}(x)$ instead of x directly. Almashaqbeh et al. [ABT24] show that a modified NY filter that stores the secret key sk as part of its internal representation M is still resilient under the AB test. However, such a construction will not satisfy Filić correctness as the Filić model allows oracle access to the internal representation M, allowing the adversary to read the secret key. The same argument can be used to show that resilience under the BP test also does not imply Filić correctness.

6 PRF-backed Standard Bloom filters

In this section, we solve Problem 5 by proving that a PRF-backed Standard Bloom filter construction is *not* strongly resilient under the BP-test. We actually prove a stronger result showing that the construction is not strongly resilient even when using truly random functions instead of PRFs.

When encoding a set, **MB** may get *saturated*, i.e, every bit in **MB**'s representation $M \in \{0,1\}^m$ is set to 1. Let p_s be the saturation probability of **MB**. Finding p_s is equivalent to solving the Coupon Collector's Problem [GKL15]. The probability of a given bit being 0 is $(1 - \frac{1}{m})^{nk}$. The probability of at least 1 of m bits being 0 is $\leq m(1 - \frac{1}{m})^{nk} \leq me^{-nk/m}$, using a union bound and the inequality $1 + x \leq e^x$ for any $x \in \mathbb{R}$ [MU17]. The saturation probability of **MB** is then $p_s \geq 1 - me^{-nk/m}$.

Theorem 5. Let **MB** be a modified construction of a Standard Bloom filter **SB** that replaces each hash function h_i in **SB** with a truly random function f_i. Let $\varepsilon \in (0,1)$ and $n \in \mathbb{N}$. **MB** is not (n,t,δ)-resilient under the BP-test for any $t \in \mathbb{N}$ and any $\delta \in (0,1)$ such that $\delta < p_s$, where p_s is **MB**'s saturation probability.

Proof. Suppose adversary $\mathcal{A}$ plays $\texttt{AdaptiveGame}_{\mathcal{A},t}(\lambda)$ using the following strategy. $\mathcal{A}$ chooses any $S \subset U$ of cardinality n and chooses t elements $x_i \leftarrow\!\!\$\, U \setminus S$. In the game, $\mathcal{A}$ uses its t allowed queries to **MB** to query each x_i. If all x_is are false positives, $\mathcal{A}$ chooses to bet, and bets on an $x^* \leftarrow\!\!\$\, U$ (We don't sample $x^* \leftarrow\!\!\$\, U \setminus (S \cup \{x_1, \cdots, x_t\})$ here as the proof is simplified when x^* is chosen independently of all x_i, and assuming U is large the probability of x^* not being distinct from all x_i is negligible). Otherwise, $\mathcal{A}$ passes. We show that the expected value of $\mathcal{A}$'s profit $C_{\mathcal{A}}$ is not negligible with this strategy. W.l.o.g., fix the size of the bit array m, the number of PRFs k, and the cardinality n of the encoded set.

Let p_{fp} be the false positive probability of **MB**, in expectation. Let $\mathrm{FP}(x^*)$ denote whether or not x^* is a false positive, E_b be the event denoting $\mathcal{A}$ betting (instead of passing), and E_s be the event denoting the saturation of **MB**. If $\mathcal{A}$ follows the given strategy, then

$$
\begin{aligned}
\mathbb{E}[C_{\mathcal{A}}] &= \frac{1}{\delta}\Pr[\mathrm{FP}(x^*) \cap E_b] - \frac{1}{1-\delta}\Pr[\neg\mathrm{FP}(x^*) \cap E_b] + 0 \cdot \Pr[\neg E_b] \\
&= \frac{1}{\delta}\Pr[\mathrm{FP}(x^*) \mid E_b]\Pr[E_b] - \frac{1}{1-\delta}\Pr[\neg\mathrm{FP}(x^*) \mid E_b]\Pr[E_b] \\
&= \Pr[E_b]\left(\frac{1}{\delta}\Pr[\mathrm{FP}(x^*) \mid E_b] - \frac{1}{1-\delta}\Pr[\neg\mathrm{FP}(x^*) \mid E_b]\right) \\
&= \Pr[E_b]\left(\frac{1}{\delta}\frac{\Pr[E_b \mid \mathrm{FP}(x^*)]\Pr[\mathrm{FP}(x^*)]}{\Pr[E_b]} - \frac{1}{1-\delta}\frac{\Pr[E_b \mid \neg\mathrm{FP}(x^*)]\Pr[\neg\mathrm{FP}(x^*)]}{\Pr[E_b]}\right) \\
&= \frac{1}{\delta}\Pr[E_b \mid \mathrm{FP}(x^*)]\Pr[\mathrm{FP}(x^*)] - \frac{1}{1-\delta}\Pr[E_b \mid \neg\mathrm{FP}(x^*)]\Pr[\neg\mathrm{FP}(x^*)]
\end{aligned}
$$

To evaluate the overall bound, we first derive expressions for $\Pr[\mathrm{FP}(x^*]$ and $\Pr[\neg\mathrm{FP}(x^*)]$, which will be needed in later calculations. Since E_s and $\neg E_s$ are collectively exhaustive events, we can use the law of total probability.

$$\begin{aligned}\Pr[\text{FP}(x^*)] &= \Pr[\text{FP}(x^*) \mid E_s]\Pr[E_s] + \Pr[\text{FP}(x^*) \mid \neg E_s]\Pr[\neg E_s] \\ &= 1 \cdot p_s + p_{fp}(1-p_s) = p_s + p_{fp}(1-p_s)\end{aligned}$$

$$\begin{aligned}\Pr[\neg\text{FP}(x^*)] &= \Pr[\neg\text{FP}(x^*) \mid E_s]\Pr[E_s] + \Pr[\neg\text{FP}(x^*) \mid \neg E_s]\Pr[\neg E_s] \\ &= 0 \cdot p_s + (1-p_{fp})(1-p_s) = (1-p_{fp})(1-p_s)\end{aligned}$$

Since each x_i is chosen uniformly randomly independent of x^*, and $\mathcal{A}$'s betting decision E_b depends only on the x_is, E_b and $\text{FP}(x^*)$ are independent events.

$$\Pr[E_b \mid \text{FP}(x^*)] = \Pr[E_b \mid \neg\text{FP}(x^*)] = \Pr[E_b]$$

$\mathcal{A}$ bets when all t uniformly randomly chosen x_is are false positive, which happens with probability 1 if **MB** is saturated and probability p_{fp}^t when **MB** is unsaturated.

$$\Pr[E_b] = \Pr[E_b \mid E_s]\Pr[E_s] + \Pr[E_b \mid \neg E_s]\Pr[\neg E_s] = p_s + p_{fp}^t(1-p_s)$$

The overall expression for $\mathbb{E}[C_{\mathcal{A}}]$ is then

$$\begin{aligned}\mathbb{E}[C_{\mathcal{A}}] &= \frac{1}{\delta}\Pr[E_b \mid \text{FP}(x^*)]\Pr[\text{FP}(x^*)] - \frac{1}{1-\delta}\Pr[E_b \mid \neg\text{FP}(x^*)]\Pr[\neg\text{FP}(x^*)] \\ &= \frac{1}{\delta}\Pr[E_b]\Pr[\text{FP}(x^*)] - \frac{1}{1-\delta}\Pr[E_b]\Pr[\neg\text{FP}(x^*)] \\ &= \Pr[E_b]\left(\frac{1}{\delta}\Pr[\text{FP}(x^*)] - \frac{1}{1-\delta}\Pr[\neg\text{FP}(x^*)]\right) \\ &= (p_s + p_{fp}^t(1-p_s))\left(\frac{1}{\delta}(p_s + p_{fp}(1-p_s)) - \frac{1}{1-\delta}((1-p_{fp})(1-p_s))\right)\end{aligned}$$

Since $p_{fp}, p_s \in (0,1)$, we can set $p_{fp} = 0$ to get,

$$\mathbb{E}[C_{\mathcal{A}}] \geq p_s\left(\frac{1}{\delta}p_s - \frac{1}{1-\delta}(1-p_s)\right) = \frac{1}{\delta}p_s^2 - \frac{1}{1-\delta}p_s(1-p_s)$$

The condition for this lower bound to be strictly positive is

$$\frac{1}{\delta}p_s^2 - \frac{1}{1-\delta}p_s(1-p_s) > 0 \implies \frac{1}{\delta}p_s^2 > \frac{p_s(1-p_s)}{1-\delta}$$

Since $p_s > 0$, we can divide by p_s, to get $p_s(1-\delta) > \delta(1-p_s)$ which is true when $p_s > \delta$. This proves that **MB** is not (n, t, δ)-resilient under the BP-test for any $\delta < p_s$, which is the statement of the theorem. □

This result shows that even replacing hash functions with ideal PRFs or random functions does not prevent the BP-test attack. The attack exploits the saturation of the bit array. If every query returns 1, the adversary can bet with non-negligible expected profit. The condition $\delta < p_s$ holds for a large number of non-trivial Standard Bloom filters used in practice. Since $p_s \geq 1 - me^{-nk/m}$, if $\delta < 1 - me^{-nk/m}$ then $\delta < p_s$. $\delta < 1 - me^{-nk/m}$ is equivalent to $me^{-nk/m} > 1 - \delta$. This evaluates to $nk > m \ln \frac{1-\delta}{m}$. A common method to approximate (but not calculate exactly [BGK+08]) the optimal number of hash functions, k, in a Standard Bloom filter is $k = \frac{m}{n}\ln 2$, as analyzed in [BM04]. Using this expression for k, the bound for δ becomes $n\frac{m}{n}\ln 2 > m \ln \frac{1-\delta}{m}$ which is $2m > 1 - \delta$ or more simply $\delta > 1 - 2m$. Since any non-trivial Standard Bloom filter uses at least 1 bit, we can assume $m \geq 1$. This simplifies the bound to $\delta > -1$, which is always true since $\delta \in (0,1)$. Thus if we apply the $k = \frac{m}{n}\ln 2$ approximation, **MB** is not (n, t, δ)-resilient under the BP test for any $t \in \mathbb{N}$ and any $\delta \in (0,1)$.

7 Conclusion and Future Work

This work advances the theory of adversarially robust Bloom filters by solving three open problems and clarifying the structure of the space. We presented the first Bloom filter constructions satisfying differential privacy guarantees, both symmetric and asymmetric, without altering query semantics. We established the first provable reduction between the simulator-based model of Filić et al. and the game-based model of Naor et al., showing that Filić correctness implies AB-test resilience. We also resolved a key open problem by proving that PRF-backed Standard Bloom filters are not resilient to the BP-test.

Our taxonomy organizes the landscape of adversarial models, robustness definitions, and privacy goals, exposing several natural but unresolved questions. In particular, we leave open whether the Filić model can be extended to learned Bloom filters, whether dynamic or repeated-query versions of the NOY tests can be defined, and how the Bender and CPS models relate to the more widely adopted NOY and Filić frameworks. We hope this work provides a foundation for developing a unified theory of privacy and robustness in probabilistic data structures.

Acknowledgements

We thank anonymous reviewers for helpful feedback and corrections. We thank Dr. Allison Bishop for helpful insights from her graduate course in Data Privacy at the City College of New York.

References

[ABT24] Ghada Almashaqbeh, Allison Bishop, and Hayder Tirmazi. Adversary resilient learned bloom filters. *arXiv preprint arXiv:2409.06556*, 2024.

[AGH70] James R Abernathy, Bernard G Greenberg, and Daniel G Horvitz. Estimates of induced abortion in urban north carolina. *Demography*, 7:19–29, 1970.

[BBL12] Giuseppe Bianchi, Lorenzo Bracciale, and Pierpaolo Loreti. " better than nothing" privacy with bloom filters: To what extent? In *International Conference on Privacy in Statistical Databases*, pages 348–363. Springer, 2012.

[BFCG+18] Michael A Bender, Martin Farach-Colton, Mayank Goswami, Rob Johnson, Samuel McCauley, and Shikha Singh. Bloom filters, adaptivity, and the dictionary problem. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 182–193. IEEE, 2018.

[BGK+08] Prosenjit Bose, Hua Guo, Evangelos Kranakis, Anil Maheshwari, Pat Morin, Jason Morrison, Michiel Smid, and Yihui Tang. On the false-positive rate of bloom filters. *Information Processing Letters*, 108(4):210–213, 2008.

[Blo70] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[BM04] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. *Internet mathematics*, 1(4):485–509, 2004.

[CPS19] David Clayton, Christopher Patton, and Thomas Shrimpton. Probabilistic data structures in adversarial environments. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2019.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[FKKU25] Mia Filić, Keran Kocher, Ella Kummer, and Anupama Unnikrishnan. Deletions and dishonesty: Probabilistic data structures in adversarial settings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 137–168. Springer, 2025.

[Fou23] Linux Foundation. Linux Kernel Documentation - BPF Maps. https://www.kernel.org/doc/html/next/bpf/maps.html, 2023. Accessed: 2023-05-02.

[FPUV22] Mia Filic, Kenneth G Paterson, Anupama Unnikrishnan, and Fernando Virdia. Adversarial correctness and privacy for probabilistic data structures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[GKL14] Thomas Gerbet, Amrit Kumar, and Cédric Lauradoux. On the (in) security of google safe browsing. *INRIA ePrint*, 2014.

[GKL15] Thomas Gerbet, Amrit Kumar, and Cédric Lauradoux. The power of evil choices in bloom filters. In *IEEE/IFIP International Conference on dependable systems and networks*, 2015.

[Goo23] Google. LevelDB Bloom Filter. https://github.com/google/leveldb/blob/main/util/bloom.cc, 2023. Accessed: 2023-05-04.

[GRW+22] Sergio Galán, Pedro Reviriego, Stefan Walzer, Alfonso Sánchez-Macian, Shanshan Liu, and Fabrizio Lombardi. On the privacy of counting bloom filters under a black-box attacker. *IEEE Transactions on Dependable and Secure Computing*, 20(5):4434–4440, 2022.

[HLM08] Joop Hox and Gerty Lensvelt-Mulders. Encyclopedia of survey research methods, 2008. Randomized Response.

[KLS+25] Yekun Ke, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Dp-bloomfilter: Securing bloom filters with differential privacy. *arXiv preprint arXiv:2502.00693*, 2025.

[LN25] Chen Lotan and Moni Naor. Adversarially robust bloom filters: Monotonicity and betting. *IACR Communications in Cryptology*, 2(1), 2025.

[Man94] Naurang S Mangat. An improved randomized response strategy. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):93–95, 1994.

[Met] Meta. RocksDB Bloom Filter. https://github.com/facebook/rocksdb/blob/main/util/dynamic_bloom.h. Accessed: 2023-05-04.

[MPR20] Michael Mitzenmacher, Salvatore Pontarelli, and Pedro Reviriego. Adaptive cuckoo filters, 2020.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[NE19] Moni Naor and Yogev Eylon. Bloom filters in adversarial environments. *ACM Transactions on Algorithms (TALG)*, 15(3):1–30, 2019.

[NO22] Moni Naor and Noa Oved. Bet-or-pass: Adversarially robust bloom filters. In *Theory of Cryptography Conference*, 2022.

[RHDS21] Pedro Reviriego, José Alberto Hernández, Zhenwei Dai, and Anshumali Shrivastava. Learned bloom filters in adversarial environments: A malicious url detection use-case. In *IEEE International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2021.

[RSMW+22] Pedro Reviriego, Alfonso Sánchez-Macian, Stefan Walzer, Elena Merino-Gómez, Shanshan Liu, and Fabrizio Lombardi. On the privacy of counting bloom filters. *IEEE Transactions on Dependable and Secure Computing*, 20(2):1488–1499, 2022.

[SBBR17] Neha Sengupta, Amitabha Bagchi, Srikanta Bedathur, and Maya Ramanath. Sampling and reconstruction using bloom filters. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1324–1337, 2017.

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[TKCY22] Shun Takagi, Fumiyuki Kato, Yang Cao, and Masatoshi Yoshikawa. Asymmetric differential privacy. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1576–1581. IEEE, 2022.

[VF24] Fernando Virdia and Mia Filić. A note on securing insertion-only cuckoo filters. *Cryptology ePrint Archive*, 2024.

[VKMK21] Kapil Vaidya, Eric Knorr, Michael Mitzenmacher, and Tim Kraska. Partitioned learned bloom filters. In *International Conference on Learning Representations*, 2021.

[War65] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American statistical association*, 60(309):63–69, 1965.

[XQZ+14] Chen Xiangyu, Du Qiaoqiao, Jin Zongda, Xu Tian, Shi Jiachen, and Gao Ge. The randomized response technique application in the survey of homosexual commercial sex among men in beijing. *Iranian journal of public health*, 43(4):416, 2014.