Differentially Private Crowd Monitoring over Encrypted Bloom Filters

Fatemeh Marzani, Thijs van Ede, Geert Heijenk, Maarten van Steen
University of Twente
Enschede, The Netherlands
f.marzani,t.s.vanede,geert.heijenk,m.r.vansteen@utwente.nl

Abstract

Accurate crowd statistics are essential for urban safety, smart city services, and event management, but also risk exposing sensitive information. Precise counts can enable adversaries to identify individuals by correlating released statistics with external data. We propose a novel protocol for privacy-preserving crowd analytics that combines Bloom filters, fully homomorphic encryption (FHE), and differential privacy (DP). Unlike prior approaches without formal guarantees, our design injects randomized DP noise directly under encryption, providing strong and quantifiable privacy protection. All computations, including aggregation and noise addition, are executed entirely in the encrypted domain, ensuring that no intermediate values are leaked. Our evaluation shows that the protocol achieves both efficiency and accuracy: relative accuracy exceeds 90% with $\epsilon = 0.5$, and aggregation of over 1000 users completes in under one second on commodity hardware. By uniting encrypted processing, formal privacy guarantees, and practical scalability, this work demonstrates the feasibility of deploying privacy-preserving crowd monitoring in real-world smart city infrastructures.

CCS Concepts

• **Security and privacy → Usability in security and privacy**.

Keywords

Privacy-preserving analytics; Crowd monitoring; Smart cities; Differential Privacy; Fully Homomorphic Encryption (FHE)

ACM Reference Format:
Fatemeh Marzani, Thijs van Ede, Geert Heijenk, Maarten van Steen. 2026. Differentially Private Crowd Monitoring over Encrypted Bloom Filters. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3748522.3779813

1 Introduction

Understanding the dynamics of crowds is important for urban planning [20], tourism management [12], and public safety [13]. In modern deployments, sensors installed in public areas collect unique identifiers, such as MAC addresses or public transport card numbers, to estimate crowd sizes at locations and infer movement between them. Crowd monitoring systems typically support two categories of queries: *footfall*, which estimates the number of unique individuals observed at a given location during a time window, and *crowd flow*, which estimates the number of people moving across locations and/or time intervals. In both cases, individual data is abstracted into aggregate statistics, but precise output can still reveal private information. Previous work has shown that only a few spatio-temporal data points are enough to uniquely identify most individuals [5]. We refer to this risk as information leakage, the ability of an adversary to infer the participation or movement of specific individuals from released statistics. This risk is especially high in small or sparse groups, where the absence or presence of a single person can significantly alter the outcome. Recent work by Marzani et al. [14] proposed a system that protects raw identifiers using encrypted Bloom filters and deterministic sampling techniques. While their architecture prevents direct access to plaintext identifiers, the noise they inject is based on deterministic patterns tied to identifiers. Such patterns can be learned or reversed over time, weakening privacy guarantees. Moreover, because noise is added at the sensor, the privacy level is fixed before aggregation and cannot be adapted at query time.

In this work, we address these limitations by introducing differential privacy (DP) [6] into the encrypted computation pipeline. Differential privacy provides a formal guarantee that the presence or absence of any single individual has only a limited effect on the published result, achieved by adding carefully calibrated random noise. In our protocol, this principle is applied directly in the encrypted domain: after aggregating encrypted Bloom filters, the server injects randomized DP noise before decryption.

Applying DP in the encrypted domain strengthens privacy in several ways. Injecting noise after aggregation prevents deterministic patterns that adversaries could learn or reverse over time, a limitation of sensor-side approaches such as [14]. Differential privacy also provides formal and quantifiable guarantees, ensuring that the presence or absence of a single individual has only a limited effect on the published result. Since noise is applied while the data remains encrypted, no intermediate values are revealed during computation. Managing noise at the aggregation point also allows flexible control of the privacy budget, enabling the system to adapt protection levels to different queries and deployment scenarios without modifying the sensors.

The workflow of our protocol is straightforward. Sensors encode locally observed identifiers into Bloom filters, encrypt them using fully homomorphic encryption (FHE), and send the ciphertexts to the server. The server, modeled as honest-but-curious, aggregates the encrypted Bloom filters to compute footfall or crowd flow statistics and injects randomized DP noise before decryption. Finally, the noisy aggregate is decrypted and released to the client. This pipeline guarantees that raw identifiers never leave the sensors,

SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779813

intermediate computations remain encrypted throughout, and the published results satisfy differential privacy.

2 Background

2.1 Bloom Filters

A Bloom filter is a probabilistic data structure used to efficiently test set membership with controlled false positive rates [1]. A Bloom filter consists of an array of m bits, all initially set to zero, and k independent hash functions. To insert an element, it is hashed by each function, and the corresponding k positions in the bit array are set to one. To test membership, the element is hashed again; if all k bits are set, membership is assumed; although false positives are possible, false negatives are not. For a desired false positive rate p and the expected number of elements inserted n, the optimal parameters m (number of bits) and k (number of hash functions) can be calculated: $m = -\frac{n \ln p}{(\ln 2)^2}$ and $k = \frac{m}{n} \ln 2$. These choices minimize the false-positive rate for a fixed filter size or, conversely, minimize the filter size for a target error rate. The number of unique elements inserted into a Bloom filter (*cardinality*, c) can be estimated by counting the number of bits set (t) to one in the filter after all insertions.

$$c = -\frac{m}{k} \ln\left(1 - \frac{t}{m}\right). \tag{1}$$

In addition to set membership, Bloom filters support approximate set operations. The *intersection* of two sets encoded as Bloom filters can be approximately computed by the bitwise AND of their bit arrays. If BF_A and BF_B are Bloom filters for sets A and B, the intersection filter $BF_{A\cap B}$ is given by

$$BF_{A\cap B} = BF_A \;\&\; BF_B, \tag{2}$$

where & denotes the bitwise AND operation. The number of set bits in $BF_{A\cap B}$ provides an estimate of $|A \cap B|$ using the same cardinality estimation formula (1). Privacy-preserving analytics allow the encoding of identifiers before encryption, ensuring that raw values are never directly exposed.

2.2 Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE) enables arbitrary computations to be performed directly on encrypted data, ensuring that no information about the underlying plaintext is revealed during processing [9]. Modern FHE schemes, such as CKKS [3], support both addition and multiplication operations on ciphertexts, making them well suited for secure data analytics. However, this capability comes at the cost of increased computational and storage overhead compared to traditional encryption.

2.3 Differential Privacy

Differential privacy (DP) provides a formal guarantee that the output of a data analysis does not reveal whether an individual is included in the dataset [6]. A randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-differential privacy if, for all datasets D and D' differing in a single record, and for all measurable sets S,

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S] + \delta. \tag{3}$$

Here, ϵ controls the privacy–utility trade-off (smaller ϵ means stronger privacy), and δ is a small failure probability.

In our setting, queries are counts of individuals. in DP expressed as a function f operating on, in our case, encrypted Bloom filters. For footfall, this is the number of unique individuals in one location, and for crowd flow, it is the size of the intersection of detected people in multiple locations. The sensitivity of f expresses the maximum change in its output when its input is changed minimally. In our case, this sensitivity is 1, since adding or removing a single individual changes the result by at most one.

To achieve ϵ-differential privacy, we apply the Laplace mechanism:

$$\mathcal{M}(D) = f(D) + \eta, \qquad \eta \sim Lap(1/\epsilon) \tag{4}$$

where η is drawn from the Laplace distribution with mean 0 and scale $1/\epsilon$, with density:

$$f_\eta(x) = \tfrac{\epsilon}{2} e^{-\epsilon|x|}$$

This mechanism is ϵ-indistinguishable: for two neighboring data sets D and D' that differ in one individual, $|f(D) - f(D')| \leq 1$, and for any output $c \in \mathbb{R}$,

$$\frac{\Pr[\mathcal{M}(D) = c]}{\Pr[\mathcal{M}(D') = c]} = \frac{f_\eta(c - f(D))}{f_\eta(c - f(D'))} \leq e^{\epsilon}.$$

Hence, the mechanism satisfies what is known as pure DP (i.e., $\delta = 0$).

In our protocol, the server never observes the plaintext counts and processes only encrypted Bloom filters. For each query, it homomorphically computes the number of set bits t and returns the encrypted value. The client applies the Bloom filter cardinality estimator (1) to obtain $f(D)$. To enforce DP, we sample $\eta \sim Lap(1/\epsilon)$ in the count domain and interpret $f(D) + \eta$ as the noisy result. Because the server operates only on encrypted set-bit totals, this noisy count is mapped back into the Bloom filter domain: we compute the number of set bits t' that correspond to η using the inverse of the cardinality formula, encrypt t', and homomorphically add it to the encrypted sum.

This ensures that noise is applied exactly as prescribed by the Laplace mechanism on individual counts, while remaining consistent with the Bloom filter estimator. All intermediate values remain encrypted: the server never learns the plaintext cardinality or number of set bits, and the client never learns the exact noise value that is added.

3 Related Work

Crowd monitoring has been widely studied using data sources such as Wi-Fi probe requests and device identifiers (e.g., MAC addresses), which support estimation of crowd densities and movement patterns [16, 19, 2]. Although effective, these methods often compromise privacy, as device identifiers can be linked, tracked, or re-identified [4]. Aggregation-based techniques, such as counting sketches [10] and k-anonymity [21], have been introduced to reduce the risk of direct identification, but remain vulnerable in low-density settings or when auxiliary data is available.

To address these risks, differential privacy (DP) has been explored in recent work. However, existing DP mechanisms fall short in practical crowd monitoring applications, to protect both footfall and crowd flow statistics. Purcell et al. [17] proposed a two-party protocol for differentially private set intersection estimation, but their approach is limited to static, pairwise scenarios and does

not support encrypted intermediate computation or multi-sensor settings. RAPPOR [8] introduces randomized noise for DP web tracking applications, using Bloom filters, but is not designed to support set intersections or crowd flow estimation. DPBloomfilter [11] applies randomized response to private membership queries, yet does not generalize to tracking movement patterns across time or space. Rusca et al. [18] proposed a system using noisy Bloom filters for crowd counting, but the added noise is canceled out during intersection operations, undermining privacy for crowd flow analysis.

A closely related system was proposed by Marzani et al. [14], they introduced Bloom filters and deterministic sampling methods for crowd monitoring. Their approach reduces precision in sparse groups to limit re-identification risks, but it does not provide formal differential privacy guarantees. Because the sampling is deterministic, the same identifiers consistently influence the output. In contrast, our protocol relies on fully homomorphic encryption with randomized Laplace noise, ensuring differential privacy at the output level.

4 System and Threat Model

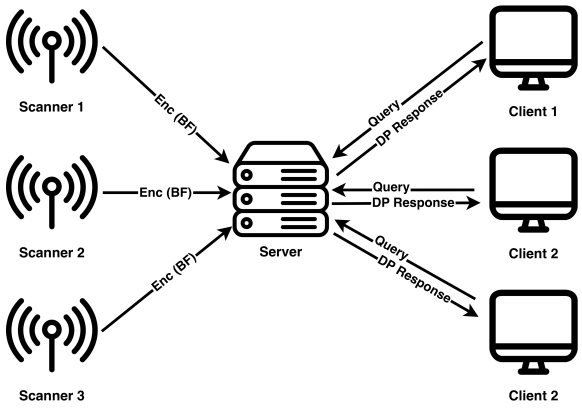


Figure 1: System Model. Scanners $S = (s_1, ..., s_n)$ collect identifiers, process these and send them to a central processing server. Clients subsequently issue footfall or crowd flow queries to estimate the number of people at given locations.

Figure 1 illustrates our architecture, which returns differentially private answers for *footfall* and *crowd flow* queries.

Data collection and encryption: In our model, we assume that each location is equipped with a logically single scanner. Such a scanner collects device identifiers observed at a location during a time interval (**epoch**), possibly through several sensors that report their data to the scanner. The scanner encodes the identifiers into a Bloom filter and encrypts this filter under the public keys of clients using fully homomorphic encryption. Scanners and their sensors are trusted to collect and encrypt data in a secure and transparent manner. We formally define footfall and crowd flow as follows:

Footfall: Let $T_{s,e}$ denote the set of detections by scanner s during epoch e. The footfall at location s for epoch e is defined as $|T_{s,e}|$.

Crowd Flow: Let $SE = \{(s_1^*, e_1^*), \ldots, (s_n^*, e_n^*)\}$ be a set of scanner-epoch pairs, with corresponding detection sets $T_{s_i^*,e_i^*}$. The crowd flow is the cardinality of the intersection, $|\bigcap_{(s_i^*,e_i^*)\in SE} T_{s_i^*,e_i^*}|$.

Encrypted aggregation and DP noise: The server receives Enc(BF) from the scanners. For footfall, it homomorphically counts the set bits in the encrypted Bloom filter to obtain $\mathsf{Enc}(t)$. For crowd flow, it homomorphically ANDs the relevant encrypted Bloom filters and then counts the set bits, again producing $\mathsf{Enc}(t)$.

To enforce differential privacy, the server does not apply the Laplace mechanism directly to the estimator in Eq. (1), as this would require evaluating a logarithm homomorphically. Instead, noise is added in the set bits domain. The server samples Laplace noise in the clear and then computes, in plaintext, the corresponding adjustment that this noise induces in the Bloom-filter bit count. This adjustment is applied under encryption using only subtraction and multiplication, avoiding expensive homomorphic logarithmic operations. The result is an encrypted noisy bit count $\mathsf{Enc}(t_{\text{noisy}})$.

Return and decryption: The server returns $\mathsf{Enc}(t_{\text{noisy}})$ to the client. The client decrypts locally and applies the Bloom filter estimator to obtain the DP-protected statistic. The server never observes plaintext identifiers, intermediate counts, or DP outputs.

4.1 Threat model

Our goal is to prevent inference and re-identification from both the computations and released results. We make the following assumptions.

- Scanners are trusted to correctly encode and encrypt observations. We do not consider these components to be physically compromised.
- The server is honest-but-curious. It executes the protocol but attempts to learn from what it sees. It never holds a private key and never observes plaintext data or intermediate results.
- Clients are honest-but-curious as well; they follow the protocol, but may try to combine the released statistics with auxiliary information to infer additional information about individuals. In our design, clients should learn only differentially private aggregate statistics.
- Concerning key management, the client owns the key pair $(\mathsf{pk}, \mathsf{sk})$. The scanners and the server receive pk; only the client holds sk and performs decryption.
- For noise generation, the server samples fresh DP noise for each query and applies its effect under encryption in the bit-count domain. This produces an encrypted noisy bit count $\mathsf{Enc}(t_{\text{noisy}})$ without revealing any intermediate plaintexts.
- We assume that there is no collusion between the server and any client. A colluding server–client pair could combine what the server knows about the applied noise with what the client decrypts, revealing the exact count and breaking the DP guarantee for that client.
- Probabilistic encryption, Bloom filters per epoch, and independent noise per query limit linkage and replay across queries.

No party, except the scanners, has access to the raw data. The server never decrypts the data, and the client decrypts only the final DP-protected statistics.

5 Methods

Algorithm 1 Privacy-Preserving Crowd Analytics Protocol

Key setup: The client generates an FHE key pair (pk, sk), keeps sk, and distributes pk to the scanners and the server.
for each scanner s **do**
  Collect identifiers observed during epoch e
  Encode identifiers as Bloom filter $BF_{s,e}$
  Encrypt: $Enc(BF_{s,e}) \leftarrow Enc_{pk}(BF_{s,e})$
  Send $Enc(BF_{s,e})$ to the server
end for
if query is footfall **then**
  $Enc(t) \leftarrow \textsc{EncryptedCount}(Enc(BF_{s,e}))$
else if query is crowd flow **then**
  $Enc(I) \leftarrow \textsc{Intersection}(\{Enc(BF_{s,e})\})$
  $Enc(t) \leftarrow \textsc{EncryptedCount}(Enc(I))$
end if
Sample Laplace noise $\eta \sim Lap(1/\epsilon)$
Map noise into Bloom filter domain: $t' \leftarrow (m - Enc(t)) \cdot \left(1 - \exp\left(-\frac{k\eta}{m}\right)\right)$
Encrypt and add noise: $Enc(t_{\text{noisy}}) \leftarrow Enc(t) + Enc_{pk}(t')$
Server sends $Enc(t_{\text{noisy}})$ to the client
Client decrypts: $t_{\text{noisy}} \leftarrow Dec_{sk}(Enc(t_{\text{noisy}}))$
Estimate cardinality: $n_{\text{hat}} \leftarrow -\frac{m}{k} \log\left(1 - \frac{t_{\text{noisy}}}{m}\right)$
Output: n_{hat}

In this section, we present the detailed design of our privacy-preserving crowd-analyst protocol, which integrates Bloom filters, fully homomorphic encryption (FHE), and differential privacy (DP). The protocol enables the secure aggregation of crowd statistics, such as footfall and crowd flow, without exposing raw data or intermediate results. The workflow consists of three phases:

(1) **Data Collection and Encryption:** Each trusted scanner encodes locally observed identifiers into a Bloom filter and encrypts them under the client's public key using FHE.
(2) **Encrypted Aggregation and Noise Addition:** The server aggregates encrypted Bloom filters, performs set operations (intersection), counts set bits, and homomorphically adds Laplace noise for differential privacy.
(3) **Decryption and Release:** The noisy encrypted result is returned to the client, who holds the private key and decrypts only the final differentially private statistic.

5.1 Data Collection

Each scanner collects all identifiers detected within a specific epoch. These identifiers are encoded into a Bloom filter BF with parameters (m, k) optimized for the expected set size and false positive rate. The Bloom filter is then encrypted using the CKKS FHE scheme, yielding $\mathsf{Enc}(\mathsf{BF})$, which is sent to the server.

5.2 Encrypted Aggregation and DP Noise

Upon receiving encrypted Bloom filters, the server computes the desired aggregates:

- For **footfall**, the server computes the encrypted sum of set bits in a single encrypted Bloom filter.
- For **crowd flow**, it homomorphically computes the intersection (bitwise AND) of the relevant encrypted Bloom filters and then counts set bits.

To enforce differential privacy, the server adds Laplace noise in the *count domain* and then maps it consistently into the Bloom filter domain:

(1) The server samples Laplace noise $\eta \sim \text{Lap}(1/\epsilon)$.
(2) The true cardinality c is estimated from t set bits using the Bloom filter estimator 1:
$$c = -\frac{m}{k} \ln\left(1 - \frac{t}{m}\right).$$
To apply the Laplace mechanism, we want the noisy count to be $c + \eta$, which should correspond to a new number of set bits $t + t'$. We therefore require
$$c + \eta = -\frac{m}{k} \ln\left(1 - \frac{t + t'}{m}\right).$$
Substituting the expression for c and simplifying:
$$\ln\left(1 - \frac{t}{m}\right) - \frac{k}{m}\eta = \ln\left(1 - \frac{t + t'}{m}\right),$$
which after exponentiation becomes
$$\left(1 - \frac{t}{m}\right) e^{-k\eta/m} = 1 - \frac{t + t'}{m}.$$
Solving for t' gives
$$t' = m - (m - t)e^{-k\eta/m} - t = (m - t)\left(1 - e^{-k\eta/m}\right).$$
The adjustment t' may be positive or negative because Laplace noise can either increase or decrease the estimated count.
(3) The server encrypts t' and adds it to the encrypted bit count:
$$\mathsf{Enc}(t_{\text{noisy}}) = \mathsf{Enc}(t) + \mathsf{Enc}(t').$$

Example. For the Bloom filter parameters $m = 9585$ and $k = 7$, we assume that the aggregation yields $t = 4800$ set bits. With $\epsilon = 0.5$, a sampled noise $\eta = -2.3$ gives
$$t' = (9585 - 4800)\left(1 - e^{-7(-2.3)/9585}\right) \approx -8.$$
The total number of noisy bits is then $t_{\text{noisy}} \approx 4792$.

The client applies the cardinality estimator 1. Without noise,
$$c = -\tfrac{9585}{7} \ln\left(1 - \tfrac{4800}{9585}\right) \approx 951.26,$$
while with noise,
$$c_{\text{noisy}} = -\tfrac{9585}{7} \ln\left(1 - \tfrac{4792}{9585}\right) \approx 948.96.$$
Thus, the result released differs approximately by -2.3, exactly matching the Laplace noise applied in the count domain. We do not compute the estimator c homomorphically on the server, add Laplace noise to it, and return the noisy estimate c to the client. The reason is that the estimator in 1 requires a logarithm, which is expensive to evaluate under FHE. To avoid this, we add noise in the bit-count domain instead of in the final estimate. This avoids costly homomorphic logarithms and keeps encrypted operations limited to subtraction, multiplication, and addition, making the protocol practical while preserving differential privacy.

This procedure ensures that the Laplace mechanism is implemented correctly on individual counts while remaining consistent with the Bloom filter estimator. The computations remain encrypted: the server never learns the plaintext values, and the client never learns the exact noise values.

5.3 Decryption and Release

The server sends $\mathrm{Enc}(t_{\mathrm{noisy}})$ to the client. Because only the client holds the private key, only the client can decrypt the noisy aggregate and apply the Bloom filter cardinality formula 1 to obtain the final statistic. At no point are raw identifiers or intermediate results revealed.

5.4 Differential Privacy Guarantee

Our mechanism satisfies pure ϵ-differential privacy by construction. As shown in Section 2.3, the Laplace mechanism ensures that adding or removing one individual changes the released result by at most a multiplicative factor e^{ϵ}. Section 5.2 demonstrated how to translate this noise consistently into the Bloom filter domain, ensuring that applying the estimator to $t + t'$ yields $f(D) + \eta$ exactly.

Importantly, this guarantee holds end-to-end in an encrypted setting. The server performs all aggregation and noise addition under homomorphic encryption and never observes the plaintext counts or intermediate values. The client decrypts only the final noisy result but does not know the exact value of the added noise. Thus, the only information revealed is the output of an ϵ-DP mechanism.

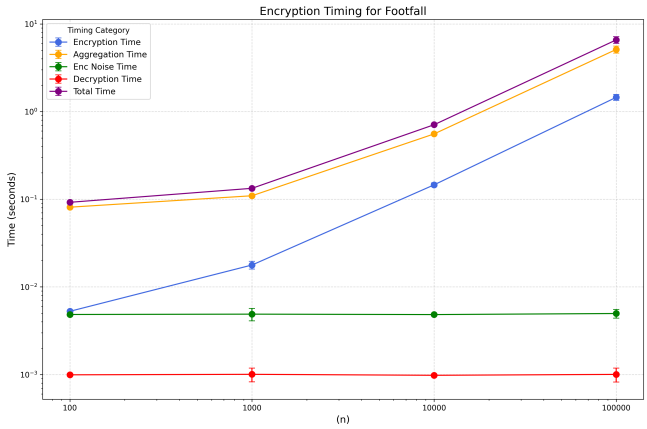
This combination of formal DP guarantees and encrypted processing ensures that both footfall and crowd flow queries are released with strong, quantifiable privacy protection, while the intermediate computations and raw identifiers remain completely hidden.

6 Evaluation

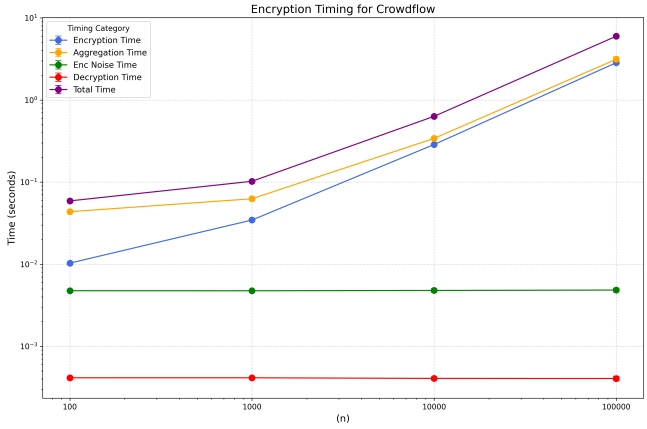
We evaluate our protocol in two dimensions: computational efficiency (timing under FHE) and privacy-utility tradeoff (statistical accuracy as a function of ϵ), for footfall (crowd size) and crowd flow (intersection) queries.[1] We use generated detections to have precise control over the size of detection sets and the number of shared identifiers between them, we generate sets of random unique identifiers to directly represent detections, mimicking pre-collected data, instead of real-world data, as they allow precise control over crowd composition. This enables us to systematically evaluate the system across a wide range of crowd flow scenarios.

6.1 Timing

We implemented the full protocol in Python using the TenSEAL CKKS library for FHE and a standard Bloom filter package. All timing experiments were performed using a MacBook Pro (M2). For timing, we varied the expected number of unique elements $n \in \{100, 1000, 10\,000, 100\,000\}$. For each n, we set the Bloom filter's target false positive rate $p = 0.01$, which determines the Bloom filter length as $m = -\frac{n \ln p}{(\ln 2)^2}$. The resulting values are $m = 958$, 9585, 95 850, and 958 505 for the four settings, respectively. The number of elements inserted per filter is $n/2$. For each n, we repeated the protocol 100 times and report the mean and standard deviation for encryption, aggregation, noise addition, decryption, and total query times. Figure 2 shows the end-to-end computation times for encrypted footfall and crowd flow queries. Each plot reports the time for each processing phase as a function of the Bloom filter size m. The results demonstrate that our protocol is efficient for practical deployments, with total query times less than a few seconds even for the largest Bloom filters tested with $m = 958\,505$.



(a) Footfall timing



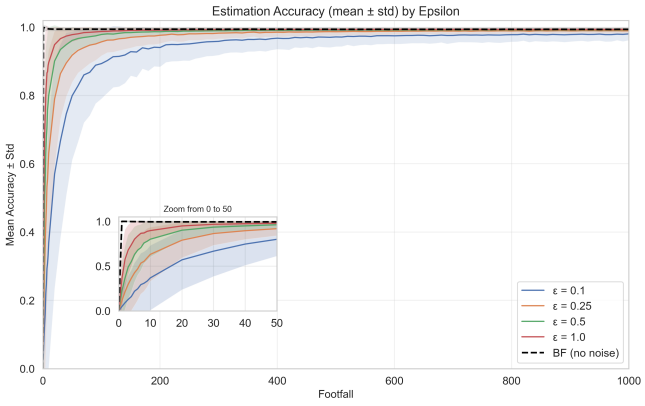
(b) Crowd flow timing

Figure 2: Computation times for (a) footfall and (b) crowd flow queries under FHE, for different Bloom filter sizes.

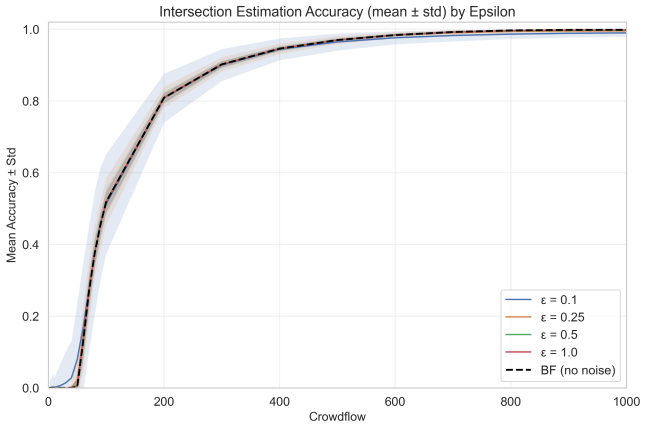
6.2 Privacy-Utility Tradeoff

To isolate the effect of DP noise on utility, we perform a separate, non-encrypted evaluation with $n = 1000$ and $p = 0.01$ and 1000 repetitions per setting. For footfall, sample sizes range from 1 to 1000; For crowd flow, both sets have 1000 random detections, and intersection sizes range from 0 to 1000. For each, we report the mean and standard deviation of estimation accuracy for privacy parameters $\epsilon \in \{0.1, 0.25, 0.5, 1.0\}$. In differential privacy literature,

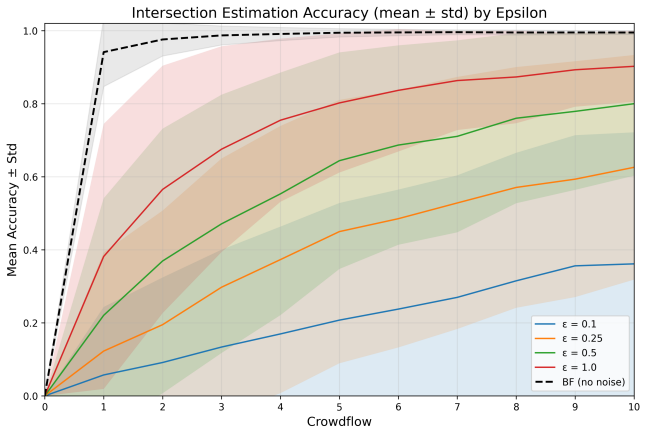
[1] The source code is publicly available. https://anonymous.4open.science/r/DP_BF_Enc-72E3



(a) Footfall accuracy



(b) Crowd flow accuracy for groups of size 1000.



(c) Crowd flow accuracy for groups of size 10.

Figure 3: Estimation accuracy (mean ± std) for differentially private queries: (a) footfall, (b) crowd flow for large groups, and (c) crowd flow for small groups, all evaluated across intersection size and privacy parameter ϵ.

smaller values of ϵ (e.g., 0.1) correspond to strong privacy guarantees, while larger values (e.g., 1.0) relax privacy but improve utility [7]. Relative accuracy is defined as:

$$\text{Relative accuracy} = \frac{\text{estimate}}{\text{actual}}$$

(with relative accuracy set to 0 if the actual value is 0).

Figure 3 shows the mean and standard deviation of estimation accuracy for both footfall and crowd flow as a function of group/intersection size and ϵ. Each curve corresponds to a different privacy parameter; the black dashed line represents the non-private Bloom filter baseline. In Figure 3 (a), we observe that for small groups, relative accuracy is limited by Laplace noise, especially for stronger privacy (lower ϵ). This trade-off is intentional and desirable; in practical scenarios, attackers can often obtain auxiliary information about individuals in small groups, which makes linkage attacks and identification far easier. By reducing the precision of published statistics for small groups, our approach mitigates these risks. As group size increases, relative accuracy rapidly approaches the non-private baseline for all values of ϵ.

Figure 3 (b) focuses on crowd flow estimation. Here, both sets A and B contain 1000 unique identifiers, and we vary the number of shared identifiers $|A \cap B|$ from 0 to 1000. The results show that the relative accuracy improves as the intersection size increases. This aligns with the behavior of Bloom filters, when more identifiers are shared, the estimation becomes more reliable [15]. However, crowd flow queries become particularly sensitive in low-density environments, such as people moving between two small shops in a quiet neighborhood or between separate entrances of a public park during early hours, where only a few individuals may pass through both locations. To simulate such sparse scenarios, we performed an additional experiment, as shown in Figure 3 (c). Here, A and B each contain only 10 identifiers, and we vary $|A \cap B|$ from 0 to 10. Without differential privacy, even a single shared detection can be inferred with high accuracy, posing a significant privacy risk. By injecting DP noise, our protocol introduces uncertainty into these estimates, preventing exposure of individuals in such small groups.

Our evaluation confirms that the proposed protocol is both practical and privacy-preserving. The FHE-based pipeline is computationally feasible for real-world deployments, achieving query times of only a few seconds even for large Bloom filters. The DP mechanism provides a tunable trade-off between privacy and utility: for moderate to large groups, the accuracy remains close to the non-private baseline across all tested values of ϵ. For small groups, where the influence of a single individual is most significant, the mechanism introduces higher relative noise. This behavior is consistent with the formal guarantee of differential privacy, which ensures that the inclusion or exclusion of any individual has only a limited impact on the released statistic.

7 Discussion

7.1 Privacy Enhancement Using Fully Homomorphic Encryption

Fully homomorphic encryption (FHE) enables confidential computations directly on encrypted data, supporting both addition and multiplication without decryption. In our protocol, all aggregation

and cardinality estimation are performed on encrypted Bloom filters, ensuring that raw data or identifiers never leave the trusted sensors. While FHE offers robust privacy, it also introduces significant computational overhead, especially as Bloom filter length m grows. To mitigate the computational overhead imposed by FHE on large Bloom filters, we implemented a segmentation approach: each Bloom filter is divided into smaller, fixed-size blocks, and each block is processed independently under encryption. This chunked processing (e.g., using 4096-bit segments) significantly reduces memory consumption and enables practical encrypted computation even for large filter sizes (up to nearly 10^6 bits in our experiments). Segmentation does not change the underlying estimator or DP mechanism; we compute the total number of set bits in a conceptual Bloom filter of length m by summing the per-segment counts. Analytically, this is equivalent to processing a single non-segmented Bloom filter. This technique allows our protocol to scale efficiently with the size of the monitored population and supports deployment in resource-constrained environments.

7.2 Trade-off Between Privacy and Utility

The choice of differential privacy parameter ϵ governs the trade-off between privacy and utility in our system. As demonstrated in our evaluation, strong privacy (low ϵ) adds substantial noise, especially affecting accuracy for small groups or sparse flows, but this is a desirable property for privacy protection. Small groups are most vulnerable to re-identification via auxiliary information, so deliberately reducing precision in these cases limits potential privacy risks. For applications where highly accurate aggregate statistics are needed and privacy requirements are less strict, larger values of ϵ can be used. Our protocol allows system designers to tune this trade-off to fit the privacy requirements of each scenario.

7.3 Identifiers and Robustness

Crowd monitoring often relies on device-based identifiers such as MAC addresses, but real-world deployments are increasingly affected by identifier randomization. Our method does not depend on the specific type of identifier used; as long as each unique entity is consistently mapped to a single Bloom filter bit pattern per epoch, our protocol yields valid estimates. However, aggressive identifier randomization can inflate crowd counts (if a device appears with multiple identifiers in one epoch) or deflate flows (if the same device changes identifier between epochs). Identifying more robust sources of identifiers, or methods for resolving randomized identifiers while preserving privacy, remains a key challenge. Notably, our protocol processes identifiers only within trusted sensors, encodes them immediately into encrypted Bloom filters, and discards them—ensuring no raw or linkable identifiers are ever transmitted or stored. All released statistics are noisy and aggregated, providing privacy by design and reducing consent requirements.

7.4 Local Differential Privacy

Local differential privacy (LDP) requires each scanner to randomize its observations before sending them to the server. Although this removes the need to trust the server to add noise, it does not work in our setting, especially for crowd flow estimation. Under LDP, each scanner perturbs identifiers or Bloom filter insertions independently. Footfall at a single location can still be estimated, although with higher variance. However, crowd flow requires the same individual to be represented in the same way across scanners. Independent randomization breaks this: the same identifier is very unlikely to map to the same Bloom filter positions at different scanners, so the bitwise AND cannot recover true overlaps. The intersection estimate collapses, and the flow statistic is not feasible. Deterministic alternatives, such as identifier-dependent sampling (e.g., Marzani et al. [14]), keep the representation consistent across scanners. However, they create stable and linkable patterns that adversaries can learn or reverse over time.

8 Conclusion

We have presented a practical protocol for privacy-preserving crowd monitoring that integrates fully homomorphic encryption (FHE), Bloom filters, and differential privacy (DP). Our design enables secure aggregation and analysis of crowd statistics, such as footfall and crowd flow, without exposing raw identifiers or intermediate computations to any untrusted party. By combining end-to-end encrypted processing with DP noise addition, our system offers formal privacy guarantees. To achieve scalability, we implemented Bloom filter segmentation, allowing efficient encrypted computation for large-scale deployments. Our experiments demonstrate that the protocol achieves a tunable trade-off between privacy and utility, and practical computation times on a personal laptop. Future work includes optimizing further for communication and computation costs, investigating alternative noise mechanisms, and evaluating the approach on real-world deployments with more complex identifier randomization or sensor settings. We believe that our framework provides a practical and secure foundation for privacy-preserving analytics in smart city and urban crowd monitoring applications.

Acknowledgements

This research is funded by the Dutch Research Council (NWO) through the PERSPECTIEF Program P21-08 'XCARCITY'. We gratefully acknowledge their financial support.

References

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

[2] Bram Bonné, Arno Barzan, Peter Quax, and Wim Lamotte. Wifipi: Involuntary tracking of visitors at mass events. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–6, 2013.

[3] {Jung Hee} Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 409–437. Springer Verlag, 2017.

[4] Mathieu Cunche, Mohamed-Ali Kaafar, and Roksana Boreli. Linking wireless devices using information contained in wi-fi probe requests. *Pervasive and Mobile Computing*, 11:56–69, 2014.

[5] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3(1):1376, 2013.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

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

[8] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 1054–1067, New York, NY, USA, 2014. Association for Computing Machinery.

[9] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[10] Michael Kamp, Christine Kopp, Michael Mock, Mario Boley, and Michael May. Privacy-preserving mobility monitoring using sketches of stationary sensor readings. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 370–386, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[11] 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.

[12] Yuan Lai and Constantine E. Kontokosta. Quantifying place: Analyzing the drivers of pedestrian activity in dense urban environments. *Landscape and Urban Planning*, 180:166–178, 2018.

[13] C. Martella, J. Li, C. Conrado, and A. Vermeeren. On current crowd management practices and the need for increased situation awareness, prediction, and intervention. *Safety Science*, 91:381–393, 2017.

[14] Fatemeh Marzani, Thijs van Ede, Geert Heijenk, and Maarten van Steen. Stop watching me! moving from data protection to privacy preservation in crowd monitoring. In *Availability, Reliability and Security*, pages 46–67, Cham, 2025. Springer Nature Switzerland.

[15] Michael Mitzenmacher. Compressed bloom filters. In *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*, PODC '01, page 144–150, New York, NY, USA, 2001. Association for Computing Machinery.

[16] A. B. M. Musa and Jakob Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, page 281–294, New York, NY, USA, 2012. Association for Computing Machinery.

[17] Michael Purcell, Yang Li, and Kee Siong Ng. Split, count, and share: A differentially private set intersection cardinality estimation protocol. In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

[18] Riccardo Rusca, Alex Carluccio, Claudio Casetti, and Paolo Giaccone. Privacy-preserving wifi-based crowd monitoring. *Transactions on Emerging Telecommunications Technologies*, 35(3):e4956, 2024.

[19] Lorenz Schauer, Martin Werner, and Philipp Marcus. Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS '14, page 171–177, Brussels, BEL, 2014. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[20] Michael Southworth. Designing the walkable city. *Journal of Urban Planning and Development*, 131(4):246–257, 2005.

[21] Valeriu-Daniel Stanciu, Maarten van Steen, Ciprian Dobre, and Andreas Peter. k-anonymous crowd flow analytics. In *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '20, page 376–385, New York, NY, USA, 2021. Association for Computing Machinery.