A Robust and Privacy Preserving Crowd Monitoring Framework: Short Paper

Raphael Ritz, Marc Damie, Thijs van Ede, and Fatemeh Marzani(✉)

University of Twente, Enschede, The Netherlands
r.ritz@student.utwente.nl, {m.f.d.damie,t.s.vanede,f.marzani}@utwente.nl

Abstract. A key aspect of crowd monitoring is quantifying crowd size. While determining the size of a crowd in a single location at a specific moment is straightforward, measuring crowd flow across locations or over extended periods requires estimating the intersection of unique individuals observed at different points in space or time. In this work, we look at the re-identification of facial data, which inherently poses significant privacy risks. We propose a privacy-preserving framework for crowd flow estimation providing both robustness against variations in images captured at different cameras (using LSH) and privacy protection to ensure biometric data is not visible outside the camera (using garbled circuits). Facial embeddings are first transformed into binary representations (LSH), ensuring that embeddings belonging to the same individual (intra-class) have low Hamming distances, while embeddings from different individuals (inter-class) are well separated. This design enables robust matching in the presence of biometric noise. To compute the crowd flow, we compare the hashes using garbled circuits, allowing collaborating camera nodes to jointly estimate crowd flow without revealing raw facial data or intermediate representations. Experimental results show that a 128-bit configuration achieves an average F1-score of 96.18% with an amortized execution time of 10.21 ms per comparison, demonstrating the practicality of the proposed approach for privacy-compliant crowd monitoring.

Keywords: Crowd monitoring · Biometric privacy · Garbled Circuits · Multiparty Computation · Locality-Sensitive Hashing · Privacy By Design

1 Introduction

Automated crowd monitoring systems play an important role in smart cities, supporting applications such as urban planning, public safety, and large-scale event management by quantifying human movement patterns. Two fundamental tasks are commonly distinguished in this domain: *footfall* and *crowd flow*. Footfall refers to counting the number of individuals observed at a single location within a given time window. Crowd flow measures how many individuals move between different locations over time [21].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
S. Nadjm-Tehrani et al. (Eds.): ARES 2026, LNCS 16898, pp. 49–60, 2027.
https://doi.org/10.1007/978-3-032-36874-4_3

While such operations are valuable in practice, crowd monitoring raises privacy concerns because access to raw observation data (e.g., camera footage) can enable mass surveillance. To address these concerns, several works have proposed privacy-preserving solutions that enable these measurement tasks without revealing raw observation data to a central party. The level of privacy risk depends strongly on the task. Footfall estimation is less privacy-sensitive because it can be computed directly from local observations. In contrast, crowd-flow estimation is inherently more privacy-sensitive because it requires linking observations made at different locations or times. This linking step introduces significant privacy risks since it enables tracking individuals across space and time.

Related Work. For *footfall* estimation, existing solutions mainly rely on camera data processed locally. Chan et al. [4] estimate crowd density using visual features without explicit tracking and report only footfall statistics. Similarly, edge-based systems such as CaCTUs [2] process visual data directly on the camera, discard raw frames, and export only aggregate statistics.

For *crowd-flow* estimation, two main approaches have emerged in the literature: re-identification and Private Set Intersection (PSI).

Camera-based re-identification techniques estimate crowd flow by matching individuals across multiple camera views. Mazzon et al. [15] and Narayan et al. [16] show that high-dimensional facial embeddings enable reliable cross-camera matching. However, these approaches rely on storing and comparing biometric representations, which raises important privacy concerns. Facial embeddings have been shown to be partially invertible, enabling reconstruction of facial images or inference of sensitive attributes [12]. This property complicates compliance with privacy regulations such as the GDPR.

Several recent solutions [13,14,18] instead rely on a cryptographic protocol called Private Set Intersection (PSI) [8]. PSI enables two parties to compute the intersection, or only its size, of two private sets without revealing the individual sets. In crowd-flow estimation, PSI is used to compute the intersection size of two sets of biometric identifiers without revealing the identifiers themselves. The size of the intersection between two cameras corresponds to the crowd flow. Stanciu et al. [18] build a PSI protocol based on homomorphic encryption and Bloom filters to compare facial embeddings. Marzani et al. [13] enhance privacy in sparsely populated regions through controlled uncertainty using sampling, and later introduce formal differential privacy guarantees in [14]. However, these solutions rely on exact matching of stable identifiers and therefore cannot support inherently noisy data sources such as facial embeddings (Table 1).

Gap in the Literature. While existing footfall estimation systems provide accurate and privacy-preserving solutions, current approaches for crowd-flow estimation still present major limitations.

Camera-based solutions [15,16] provide no privacy guarantees because they require exchanging facial embeddings from which private information can be extracted [12]. Hence, these approaches cannot be considered privacy-preserving.

PSI-based solutions provide strong privacy guarantees but rely on exact matching of biometric information. Biometric data is inherently noisy because

Table 1. Comparison of crowd monitoring systems.

Work	Task	Privacy-Preserving	Robust to Biometric Noise
Chan et al. [4]	Footfall	✔	✔
CaCTUs [2]	Footfall	✔	✔
Stanciu et al. [18]	Both	✔	✘
Marzani et al. [13,14]	Both	✔	✘
Camera Re-ID [15,16]	Both	✘	✔
This work	**Crowd-flow**	✔	✔

a face can be captured from different positions or under different lighting conditions. As a result, the extracted representations can vary between locations and time windows. Exact matching therefore leads to many false negatives and biases the estimation toward lower values. Consequently, these solutions lack robustness to biometric noise.

In summary, there is currently no crowd-flow estimation framework that provides both strong privacy guarantees and robustness to biometric noise.

Our Contributions. We introduce a crowd monitoring framework that estimates the crowd flow between two cameras in a privacy-preserving and robust manner. Our solution combines facial vector embeddings, locality-sensitive hashing (LSH), and garbled circuits (GCs). LSH is a hashing technique that is insensitive to small variations in the input. We use this property to ensure robustness to biometric noise. GCs are a cryptographic protocol that enables two parties to evaluate a function without revealing their private inputs. We use GCs to compare the hashes produced by two cameras without revealing the hashes themselves. The GC evaluation directly outputs the number of matching hashes, which corresponds to the crowd-flow estimation. The protocol ensures that no information about biometric data is revealed beyond the crowd-flow statistic in the semi-honest adversarial model.

To demonstrate the practicality of our approach, we implement[1] our protocol using the MPC4J framework and evaluate it on the FERET Color and SCFace datasets [10,17]. Our experiments reach an average F1-score of 96.18% for 128-bit hashes and an amortized execution time of 10.21 ms per comparison.

[1] The implementation of the proposed system is publicly available at: https://github.com/utwente-scs/FPSI-for-crowd-monitoring.

2 Our Crowd Monitoring Framework

System Presentation. In crowd flow monitoring, cameras are deployed at different locations, and the objective is to measure the flow of individuals between two locations. This requires comparing the images captured by two cameras in order to identify individuals observed at both sites. As in previous work [13,18], our solution relies on a two-party protocol that enables two cameras to measure the flow between their respective locations without relying on an external party. This protocol can be executed in a peer-to-peer manner between each pair of locations. The output of the two-party protocol, namely the flow measurement, can then be transmitted to a client that centralizes these statistics. With this peer-to-peer design, the images captured by the cameras never leave the devices.

Threat Model. Our two-party protocol involves two camera nodes C_A and C_B, that observe disjoint locations. We assume a semi-honest adversarial model in which both cameras follow the protocol but may attempt to infer *passively* additional information about the other party's data from the exchanged messages. We make two additional assumptions that are standard in this setting. First, a secure communication channel exists between the nodes, which can for instance be instantiated using TLS. Second, the parties do not collude, meaning that they are not simultaneously corrupted. If both parties were corrupted, no secure protocol could protect their data, since corruption already reveals the data to the adversary.

Solution Overview. Our framework consists of three main steps. First, each camera extracts biometric data from the raw image frames. This step outputs vector embeddings that encode the facial or biometric features of each individual observed at the location. Second, each camera applies locality-sensitive hashing (LSH) to transform each vector embedding into a compact binary hash. LSH is a class of hashing functions designed to preserve similarity, meaning that small variations in the input result in similar hash values. This property provides robustness to noise in the captured images. Third, the two cameras jointly execute a cryptographic protocol to compute the Hamming distance between their hashes without revealing the hashes themselves. For this purpose, we rely on Garbled Circuits (GC), a standard framework for such secure computations. Figure 1 illustrates the overall protocol. The following paragraphs describe each step in more detail.

Facial Data Embedding. This step is straightforward, as it relies on existing tools for face detection and facial embedding extraction. In our implementation, faces are detected using VOLO [20], and embeddings are computed using a ResNet-50 model pre-trained on VGGFace2 [3]. These components are modular and can be replaced with alternative detection or embeddings, provided that the resulting embeddings are high-dimensional real-valued vectors.

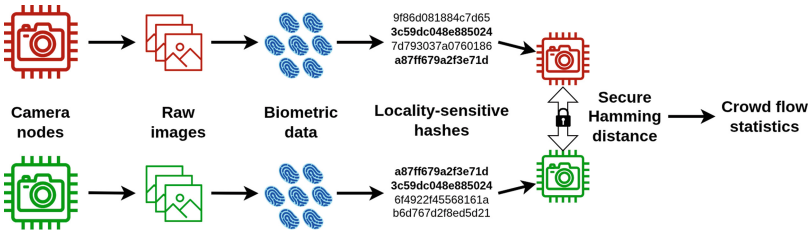


Fig. 1. High-level overview of our framework

Hashing. Facial embeddings exhibit two main limitations. First, they are high-dimensional real-valued vectors, which makes storage and pairwise comparison costly in large-scale deployments where cameras may observe thousands of individuals. Second, their real-valued structure is not well suited for most secure computation protocols, which are typically designed for binary or finite-field inputs. For this reason, directly processing real-valued embeddings within cryptographic protocols is inefficient.

To address this issue, prior work such as [18] has relied on classical hashing to obtain compact binary representations that are easy to store and compatible with secure computation. However, standard cryptographic hash functions are highly sensitive to input variations. Even a small perturbation in the facial embedding produces a completely different hash value. As a consequence, two embeddings corresponding to the same individual may be mapped to unrelated hashes, leading to false mismatches.

To achieve robustness to biometric noise, we instead rely on LSH, designed so that similar inputs are likely to produce similar hash values. More specifically, we use the scheme proposed by [5]. In this construction, each embedding is projected onto k random hyperplanes, and one bit is assigned per projection according to the sign of the dot product, that is $h_i(\mathbf{x}) = 1$ if $\mathbf{x} \cdot r_i \geq 0$, and 0 otherwise. Charikar et al. [5] show that the expected Hamming distance between two hashes is proportional to the angular distance between the corresponding embeddings. Consequently, embeddings of the same individual are likely to lie on the same side of most hyperplanes, resulting in a small Hamming distance between their binary representations. We therefore define a similarity threshold δ such that two hashes are considered to correspond to the same individual if their Hamming distance is at most δ, and to different individuals otherwise.

Secure Hamming Distance. Binary hashes are less privacy-sensitive than raw images or facial embeddings. However, an adversary may still infer private information from these hashes, in particular with LSH. To provide stronger privacy guarantees, the Hamming distance should therefore be computed using a cryptographic protocol that keeps the hashes secret and reveals only the output, namely the number of matching elements between the two sets.

To enable such secure computations between multiple parties, the cryptographic literature has developed protocols known as multi-party computations

(MPC). Among the many existing MPC constructions, we rely on one of the most established approaches: GC [19].

GC is a cryptographic protocol that enables two parties to evaluate a function $f(x_A, x_B)$ on their respective private inputs x_A and x_B. At the end of the protocol execution, only the output $f(x_A, x_B)$ is revealed, while x_A and x_B remain secret. Any function that can be expressed as a Boolean or arithmetic circuit can be evaluated using GC. In the protocol, one party acts as the garbler and the other as the evaluator. The garbler constructs a garbled version of the circuit corresponding to f and encoding its input x_A. The garbled circuit is then transmitted to the evaluator. To obtain the encoded input corresponding to x_B, the parties execute an Oblivious Transfer protocol. The evaluator can then evaluate the garbled circuit and obtain an encoded output. Finally, the parties combine their information to recover the value of $f(x_A, x_B)$. This paragraph provides only a brief GC overview and we refer the reader to [19] for a detailed description.

Complete Circuit. In our framework, each camera owns a set of hashes, so our GC implements a more complex function able to batch process sets of hashes. The function f takes as input two sets of k-bit hashes and outputs the number of elements from one set that have at least one match within Hamming distance δ in the other set. The hashes of one set are encoded as packed bit vectors and compared simultaneously against the elements of the other set, allowing many Hamming-distance computations to be performed in parallel within the circuit. The underlying circuit is composed of XOR gates, comparisons with the threshold δ, OR gates for per-element match accumulation, and an adder tree for the final match count. All these operations are easily supported by GC.

It is essential for privacy to reveal only the total number of matches and not the individual matches. Revealing individual matches would allow the cameras to know which individuals were present in both locations. Thus, the batch processing is necessary to preserve the expected privacy of the computation.

At the end of the GC evaluation, only the evaluator learns the number of similar hashes; the garbler learns nothing. This value corresponds to the crowd-flow estimate and can be transmitted to a client or a central database. The use of LSH ensures robustness to variations in image capture and feature extraction. Under the semi-honest adversarial model, GC ensures that no information about the hashes is revealed beyond the crowd-flow statistic.

Exactness of the Secure Computation. Note that our GC-based protocol implements the Hamming distance computation exactly. In other words, a plaintext (i.e., non-privacy-preserving) execution of the same Hamming distance computation would produce exactly the same output as our GC protocol. This property may appear trivial, but it is not always present in cryptographic protocols. Some privacy-preserving constructions approximate certain operations, which can lead to small differences compared to plaintext executions. In our case, the computation remains exact because LSH produces binary representations. This repre-

sentation allows the Hamming distance to be computed exactly using Boolean operations supported by GC.

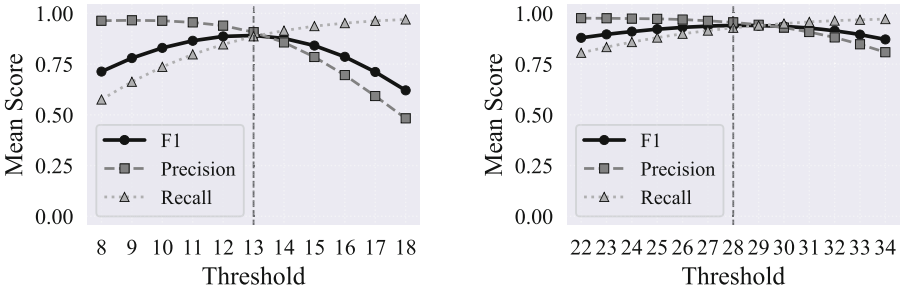
3 Evaluation

We evaluate the proposed framework in three aspects: (i) selection of the Hamming-distance threshold δ, (ii) accuracy of biometric matching, and (iii) computation and communication cost of the secure similarity protocol. The evaluation is performed on the FERET Color and SCFace datasets [10,17], where faces are detected using VOLO [20], and embedded using a ResNet-50 model.

In this experimental setup, we randomly assign images of 50 people to camera C_A and 50 people to camera C_B. To emulate cross-camera observations, embeddings of the same individual on different cameras are always selected from different camera angles, ensuring that identical embeddings never appear in both sets. Within the experiments, we vary the number of individuals appearing in both sets from 0 to 50, conducting 5 trials at each level (255 trials in total). The experimental pipeline consists of two stages: threshold selection followed by evaluation of the secure protocol implemented in Java using MPC4J.

Threshold Selection. To understand if two LSH hashes represent the same individual, we compute the Hamming distance d_H between these representations and check if this distance is lower than a threshold, i.e., $d_H \leq \delta$. To determine δ, we split the FERET dataset into 20% for threshold selection and 80% for the remaining evaluation. Of this threshold selection dataset, we randomly sample two sets of 50 identities representing observations at C_A and C_B. The number of identities appearing in both sets defines the ground-truth intersection size $GT \in [0, 50]$. Facial embeddings are transformed into binary hashes using LSH with k random hyperplanes ($k = 64$ and $k = 128$). For each candidate threshold δ, we compare the detected matches with the ground-truth correspondences and compute the corresponding precision, recall, and F1-score. The experiment is repeated 100 times for each value of GT to obtain stable averages. Figure 2 reports the resulting mean precision, recall, and F1-score as a function of the threshold. The F1-score reaches its maximum at $\delta = 13$ for $k = 64$ and at $\delta = 28$ for $k = 128$. These values provide the best balance between precision and recall and are therefore selected for the subsequent experiments.

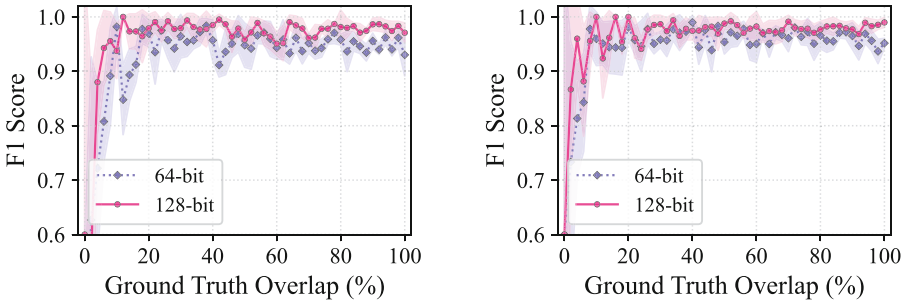
Secure Crowd-Flow Simulation. We simulate two camera nodes C_A and C_B observing different locations, each receiving a set of facial embeddings. The embeddings are transformed into binary hashes using LSH with $k = 64$ and $k = 128$ random hyperplanes. Two camera instances are created: C_A, responsible for constructing the garbled circuit, and C_B, responsible for executing it. The circuit computes pairwise Hamming distances and checks whether $d_H \leq \delta$. The output is the total number of matching embeddings observed at both C_A and C_B, thereby not leaking additional information. The estimated crowd-flow statistics is obtained by counting embeddings from C_A that have at least one match in C_B, counting multiple matches only once to avoid overestimation. Figure 3 shows the resulting matching performance (F1-score) as a function of the ground-truth



(a) $k = 64$ (b) $k = 128$

Fig. 2. Threshold selection for LSH with $k = 64$ and $k = 128$.

overlap for both datasets. Furthermore, Table 2 shows the average matching performance in terms of precision, recall and F1-score. Here we find that increasing the hash length from $k = 64$ to $k = 128$ consistently improves performance.



(a) SCFace dataset (b) FERET dataset

Fig. 3. Matching accuracy (F1-score) for $k = 64$ and $k = 128$ on SCFace and FERET.

Protocol Costs. Table 2 reports the average matching accuracy and protocol overhead. Computation time reflects the cost of generating and evaluating the garbled circuit, while communication time measures the data exchanged between the parties. The communication cost (KB) denotes the total amount of data transmitted per comparison.

As explained in Sect. 2, our GC processes two sets of hashes at a time. The cost of the overall protocol is therefore proportional to the sizes of the two sets. For clarity, we report amortized costs obtained by dividing the total protocol cost by the number of comparisons (i.e., pairs of hashes). These amortized values are easier to interpret and to extrapolate to real-world deployments.

Increasing the hash length improves accuracy but increases both computation and communication costs due to larger GCs.

Table 2. Average matching performance and protocol cost per comparison.

Dataset	k	Performance			Protocol Cost (ms)		
		Precision	Recall	F1	Computation	Communication	Total*
FERET	64	96.0%	94.8%	94.5%	2.06	2.90 (9.2KB)	5.00
	128	97.7%	96.6%	96.5%	4.17	5.91 (19.0KB)	10.12
SCFace	64	94.8%	92.4%	92.0%	2.09	2.92 (9.2KB)	5.06
	128	97.0%	96.0%	95.8%	4.26	5.99 (19.0KB)	10.29

*Total includes minor system and initialization overhead

4 Extensions

The experimental results presented in Sect. 3 show that our framework provides robustness and privacy while maintaining high efficiency. These findings indicate that the combination of LSH and garbled circuits (GC) is a promising direction for designing trustworthy crowd monitoring systems. To further advance this research direction, we identify three possible research directions.

Security against Malicious Parties. Our threat model assumes semi-honest parties, meaning that the cameras follow the prescribed protocol but may attempt to infer private information passively. This assumption is a common starting point in the secure computation literature. However, it naturally leads to a stronger security notion, namely security against malicious parties that may deviate arbitrarily from the protocol in order to reconstruct private input data.

Designing a variant of our framework that achieves malicious security is therefore a relevant direction for future work. Our GC construction relies only on basic arithmetic and Boolean operations, namely XOR, addition, and comparison. For such operations, the cryptographic literature already provides GC protocols with malicious security guarantees [11]. Extending our framework to the malicious setting would therefore require re-implementing the Hamming distance computation using recent improvements of Yao's GC that ensure security against malicious adversaries.

Enhancing the Output Privacy. Our framework ensures that the crowd monitoring process does not reveal information about the inputs. However, the monitoring output itself may still leak sensitive information. For example, if the matching procedure is executed between two cameras and one camera holds only a single image, a positive match would imply with certainty that a specific individual appeared at both locations. Since the objective of crowd monitoring is to estimate population flow rather than to track individuals, it is desirable to further mitigate such risks. A simple mitigation strategy is to allow matching between two cameras only if each holds a sufficiently large batch of observations. Enforcing a minimum batch size prevents the extreme scenario described above. In the presence of malicious adversaries, compliance with this requirement could be enforced using cryptographic mechanisms such as zero-knowledge proofs [9]. A more formal mitigation can be achieved using Differential Privacy (DP) [7]. This statistical framework requires adding calibrated noise to computation results in

order to limit the influence of any single individual. More precisely, DP ensures that the distribution of the output remains statistically similar whether or not a specific (input) data point is included. Consequently, the presence of a particular individual cannot be reliably inferred from the released crowd flow statistics.

Three-Party Protocols. Our framework relies on garbled circuits to implement a two-party secure computation. However, the cryptographic literature shows that three-party protocols can achieve significantly lower communication and computation costs. For example, a vector dot product can be computed with $O(1)$ communication complexity in certain three-party protocols, whereas two-party constructions typically require $O(n)$ communication [6]. Introducing a third party in our secure Hamming distance protocol could therefore enable the use of more efficient constructions.

More generally, this observation indicates that exploring alternative threat models may lead to more efficient variants of our framework. Future work could also investigate cryptographic primitives such as (Fuzzy) Private Set Intersection [1] as an alternative to our secure Hamming distance protocol.

5 Conclusion

Crowd-flow estimation is important for many smart-city applications but raises significant privacy concerns because it requires linking observations of individuals across locations. In this work, we proposed a privacy-preserving framework that enables robust matching of detected faces without revealing sensitive information. Our approach combines locality-sensitive hashing with garbled circuits to support robust biometric matching while revealing only the final crowd-flow statistic. Experiments on the FERET and SCFace datasets show that the approach achieves high matching accuracy with practical performance. In particular, the 128-bit configuration reaches F1-scores above 96% while requiring only about 10 ms per comparison, demonstrating that privacy-preserving and efficient crowd-flow estimation can be achieved.

Acknowledgements. The authors have no competing interests to declare that are relevant to the content of this article. This work was supported by Netherlands National Science Foundation (NWO) through the PERSPECTIEF Program P21-08 'XCARCITY'

References

1. van Baarsen, A., Pu, S.: Fuzzy private set intersection with large hyperballs. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 340–369. Springer (2024)
2. Beugin, Y., et al.: Building a privacy-preserving smart camera system (Jan 2022). arXiv:2201.09338 [cs]
3. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: a dataset for recognising faces across pose and age . In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 67–74. IEEE Computer Society, Los Alamitos (2018)
4. Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: counting people without people models or tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (Jun 2008), iSSN: 1063-6919
5. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the thirty-fourth annual ACM symposium on Theory of computing (STOC) (2002)
6. Damie, M., Hahn, F., Peter, A., Ramon, J.: Secure sparse matrix multiplications and their applications to privacy-preserving machine learning
7. Dwork, C.: Differential privacy. In: International Colloquium on Automata, Languages, and Programming, pp. 1–12. Springer (2006)
8. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Kanade, T., et al. (eds.) Advances in cryptology - EUROCRYPT 2004, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
9. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptol. **7**(1), 1–32 (1994)
10. Grgic, M., Delac, K., Grgic, S.: Scface - surveillance cameras face database. Multimed. Tools Appl. **51**(3), 863–879 (2011)
11. Lindell, Y., Pinkas, B.: Secure two-party computation via cut-and-choose oblivious transfer. J. Cryptol. **25**(4), 680–722 (2012)
12. Mai, G., Cao, K., Yuen, P.C., Jain, A.K.: On the reconstruction of face images from deep face embeddings. In: IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI) (2019)
13. Marzani, F., van Ede, T., Heijenk, G., van Steen, M.: Stop watching me! moving from data protection to privacy preservation in crowd monitoring. In: Dalla Preda, M., Schrittwieser, S., Naessens, V., De Sutter, B. (eds.) Availability, Reliability and Security, pp. 46–67. Springer Nature Switzerland, Cham (2025)
14. Marzani, F., van Ede, T., Heijenk, G., van Steen, M.: Differentially private crowd monitoring over encrypted bloom filters. In: Proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), pp. 1–8. Association for Computing Machinery (2026)
15. Mazzon, R., Tahir, S.F., Cavallaro, A.: Person re-identification in crowd. Pattern Recogn. Lett. **33**(14), 1828–1837 (2012)
16. Narayan, N., Sankaran, N., Setlur, S., Govindaraju, V.: Re-identification for online person tracking by modeling space-time continuum. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1519–151909. IEEE, Salt Lake City, UT, USA (2018)
17. Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The feret evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10), 1090–1104 (2000)

18. Stanciu, V.D., Steen, M.v., Dobre, C., Peter, A.: Privacy-preserving crowd-monitoring using bloom filters and homomorphic encryption. In: Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, EdgeSys '21, pp. 37–42. Association for Computing Machinery, New York, April 2021
19. Yao, A.C.C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), pp. 162–167 (1986)
20. Yuan, L., Hou, Q., Jiang, Z., Feng, J., Yan, S.: Volo: Vision outlooker for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **45**(5), 6575–6586 (2023)
21. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.Q.: Crowd analysis: a survey. Mach. Vis. Appl. **19**(5–6), 345–357 (2008)