SoK: Evaluation of Methods for Privacy Preserving Edge Video Analytics

Arun Joseph[0009−0009−7357−1473] and Vinod Ganapathy[0000−0003−3001−0800]

Indian Institute of Science, Bangalore, India
{arunj,vg}@iisc.ac.in

Abstract. Collaborative edge data analytics is gaining prominence as vast amounts of data are increasingly generated at the network edge by smartphones, sensors, and smart cameras. In this work, we focus on the challenge of privacy-preserving video analytics on edge devices. Video data presents unique difficulties compared to other data types, primarily due to its large size and the nature of video analysis tasks. We evaluate and compare two leading techniques for enabling secure collaborative video analytics in distributed edge environments: Secure Multiparty Computation (MPC) and Trusted Execution Environments (TEE). To assess their effectiveness, we implement five real-world case studies, including object re-identification, scene similarity detection, vehicle counting, and machine learning–based video fusion tasks. Additionally, we benchmark fundamental image processing operations under various MPC configurations to identify the most efficient MPC protocols for video workloads. Our results show that while TEE offer significant performance benefits, especially for machine learning intensive tasks, MPC remains a practical alternative in scenarios without trusted hardware, particularly when using optimized secret sharing based 3-party protocols. We provide a comprehensive analysis of performance, security, and implementation complexity for both approaches.

Keywords: Secure Edge computing · Joint video analytics · Secure multiparty computations · Trusted execution environments.

1 Introduction

The rise of IoT devices, smartphones, smart cameras, and edge sensors has led to a surge in data generation at the network edge [56, 52]. This data often includes sensitive information like health records, financial transactions, and private videos, raising significant privacy concerns. Privacy-preserving collaborative analytics at the edge addresses these concerns by keeping data local, thereby enhancing privacy, scalability, and cost-effectiveness [1, 54]. Edge-based data analytics enables real-world applications such as analyzing traffic or utility data from multiple cameras without exposing sensitive footage [59], supporting secure medical diagnoses [18], or sharing insights across factories while protecting proprietary data [68]. Several studies have explored privacy-preserving machine learning for edge collaboration [44, 60, 45, 48], and systems like [47, 25] facilitate secure edge analytics. PERQS [29] further supports privacy-preserving queries on distributed CCTV video streams using edge-based processing.

In this work, we address the problem of privacy-preserving video analytics on edge devices. Video data poses unique challenges compared to other data types, primarily due to (a) its large size, which makes communication and computation significantly more resource intensive, and (b) the nature of video analytics tasks, which are often fuzzy and statistical rather than discrete and deterministic. Tasks such as object detection or boundary identification typically rely on complex neural network models rather than rule-based algorithms, making secure processing more difficult. We evaluate and compare two prominent approaches for enabling secure collaborative video analytics at the edge: Secure Multiparty Computation (MPC) [13, 75] and hardware-based Trusted Execution Environments (TEE) [46, 61].

MPC is a cryptographic approach that enables multiple parties to collaboratively compute a function over their inputs while keeping those inputs private. This method is entirely software-driven and does not require specialized hardware, making it versatile and applicable across diverse edge devices. Several previous works have used MPC for privacy-preserving data analytics, demonstrating its ability to securely process sensitive data without exposing raw inputs. On the other hand, TEE relies on secure hardware components to create isolated environments where computations can be performed securely, even if the host system is compromised. TEE has challenges, including hardware dependency, potential vulnerabilities due to side-channel attacks, and scalability limitations in highly distributed edge systems.

Both approaches are viable solutions for enabling secure and private computation in distributed environments. Each method has distinct strengths and limitations regarding performance, scalability, deployment complexity, and security guarantees. For example, MPC excels in settings with high distrust among parties but may struggle in low-bandwidth environments. Conversely, TEE provides high computational efficiency but requires specialized hardware and may face adoption challenges due to hardware costs or vendor lock-in. In addition, MPC protocols offer a variety of design choices, including in-house vs. outsourced computation models, arithmetic vs. boolean, optimizations for 2 or 3-party settings, and different adversary models(active/passive). With recent advances and optimizations, their relative advantages, disadvantages, and suitability for privacy-preserving edge data analytics remain unclear. The primary contribution of this paper is to evaluate the relative merits of these two methods by analyzing their practical implementations and real-world performance of joint video analytics in edge computing environments. We will systematically assess the pros and cons of MPC and TEE across various metrics. The criteria on which we wish to evaluate the methods are:

- What are the computational and communication overhead? Performance evaluations to find which method is suitable for each application.
- What are the security implications of these methods? Strengths and vulnerabilities in protecting private data.
- What are the additional requirements to implement these solutions? Additional hardware or software required.
- How much effort is required to implement the solution? Ease of integration and deployment in edge devices.

By providing in-depth comparison and evaluation, we aim to clarify the trade-offs between MPC and TEE for privacy-preserving edge analytics and offer insights into which approach is better suited for different applications or constraints. This study will contribute to the advancement of secure edge analytics and help guide future developments in privacy-preserving technologies.

2 Background

Privacy-preserving data analytics is crucial as the amount of sensitive data generated by edge systems grows. This section introduces MPC and TEE and their design choices.

2.1 Secure Multiparty Computation (MPC)

Secure Multiparty Computation (MPC) [13, 75, 30, 19] is a cryptographic protocol that enables multiple parties to collaboratively compute a function over their private inputs while ensuring that those inputs remain confidential. No information about the inputs is disclosed beyond what can be inferred from the final output of the computation. Unlike other privacy-preserving approaches, MPC operates without relying on a central trusted authority. Instead, the computation is distributed among the participating parties, and the protocols are designed to be resilient even in the presence of malicious actors who may attempt to disrupt the process or extract private information.

MPC Protocols vary based on security models, defending against semi-honest and malicious adversaries. Private inputs can be securely shared and computed using secret sharing and garbled circuits. Computation can be performed in-house or outsourced, with optimizations tailored for 2-party, or 3-party, or multi-party setups. While many variations have been explored and improved over the years, it remains unclear which approach best suits privacy-preserving video analytics. Some of the key ideas used in the design of a MPC protocol are:

① **Adversary Type (Active/Passive):** The Semi-honest(Passive) [20, 30] setting assumes that all participants follow the protocol but may attempt to infer private information from the data they receive. Provides weaker security guarantees but, in general is more efficient in terms of computation and communication overhead. In contrast, malicious(Active) [41, 30] setting assumes that participants may deviate from the protocol, alter computations, or attempt to manipulate results to gain unauthorized information. It is more secure but has higher computational and communication overhead because of the additional security layers to protect against the malicious adversary.

② **Computation Domain (Arithmetic/Boolean):** Arithmetic MPC protocols (operating modulo a prime or power of two) [40] is more efficient for mathematical computations and real-world numerical applications. Numbers are split among participants and computed securely. Boolean/Binary MPC protocols are better suited for logic-based operations but incurs higher communication overhead. Often implemented using garbled circuits and Oblivious Transfer (OT) [20].

③ **Computation Choice (Secret Sharing/Garbled Circuits):** In Secret Sharing [11] techniques, inputs are divided into "shares" distributed among the participants. Each share is meaningless but can reconstruct the input when combined with other shares. This ensures that no single party has access to the entire input. While Garbled Circuits [6, 75] represent a function as a circuit with encrypted values assigned to inputs and intermediate computations, ensuring no private data is exposed in the computation.

④ **Number of Parties (2-party/3-party/multi-party) :** Generally, 2-party setting [20] requires stronger cryptographic techniques since both parties must ensure privacy with-

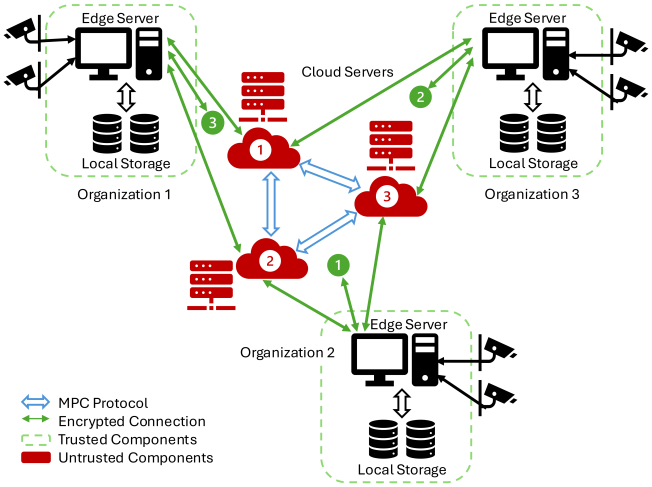

Fig. 1. MPC outsourced model where secret shares of input is encrypted and send to the cloud servers. Then the servers will perform an MPC computation to evaluate the function on the secret input. This example is of a 3-party outsourced computation includes 3 independent servers.

out a third-party mediator. Contrary 3-party setup [37] can leverage an honest majority for efficiency and security improvements. In the case of 4 or more parties typically, MPC protocols are inefficient as number of parties increases [19, 20].

⑤ **In-house/Outsourced:** In in-house MPC [37] all participating parties execute the MPC protocol on their infrastructure without relying on external computing resources. Which is limited by the available hardware and network capacity of participating entities. In outsourced MPC model [17] major part of the computation is securely offloaded to external cloud servers. It is best suited for scenarios where participants have limited computational power, such as edge devices. Figure 1 illustrates three server outsourced MPC execution model.

MPC protocols integrate well with distributed systems and edge computing, enabling secure and privacy-preserving computations in various applications [37, 17, 40]. However, MPC faces several challenges; need for continuous communication between parties increases bandwidth usage and computational overhead, especially in large-scale distributed systems. Additionally, reliance on iterative cryptographic operations can introduce latency, making it slower than other privacy-preserving methods. Optimizations and hybrid solutions can help overcome these limitations and broaden its adoption.

2.2 Trusted Execution Environment (TEE)

A Trusted Execution Environment (TEE) [46] is a secure enclave within a processor designed to protect sensitive data and code, ensuring their confidentiality and integrity. TEE provides an isolated execution environment where trusted applications can operate securely, independent of the operating system and any potentially malicious software on the device. This hardware-enforced isolation protects sensitive computations and data from unauthorized access or tampering, even if the central system or OS is

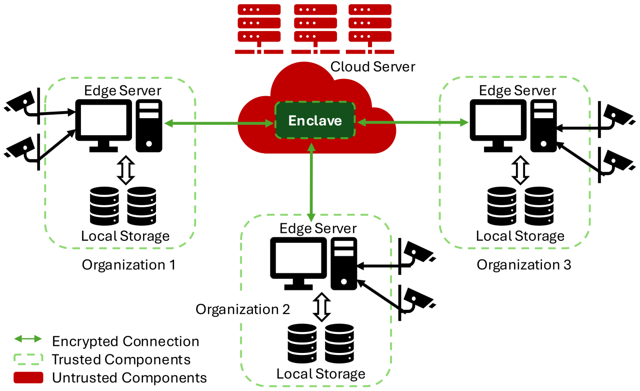

Fig. 2. Three organizations perform collaborative data analytics using a trusted execution environment (TEE) available in a cloud server.

compromised. TEE hardware can be available locally or on a remote server. Figure 2 shows edge servers utilizing a cloud-based secure enclave to perform joint computations. Some of the commonly available TEE implementations are as follows:

- Intel Software Guard Extensions (SGX) [10]: A hardware feature that provides secure enclaves for applications, enabling sensitive computations in an isolated memory region inaccessible to other processes or the OS.
- AMD Secure Encrypted Virtualization (SEV) [58]: A virtualization-based TEE solution that encrypts virtual machine memory to protect data from unauthorized access by hypervisors or other virtual machines.
- AWS Nitro [2, 4]: A virtualization-based TEE built on top of the AWS Nitro System. Nitro enclaves are run isolated from the host EC2 instance, and external networking, storage, or interactive access is not allowed.
- ARM TrustZone [3]: A system-wide approach that creates secure and normal worlds within a processor, allowing sensitive operations to run in an isolated environment.

Intel SGX creates secure enclaves for isolating small, sensitive workloads with strong security guarantees but limited memory and no system call support, making development complex. AWS Nitro Enclaves, in contrast, isolate larger workloads on EC2 instances, support integrated attestation and secure communication, and are easier to use in cloud environments. TEE use hardware-enforced isolation to protect sensitive computations from software based attacks, offering strong security for edge and cloud applications. However, they require specialized hardware, are hard to scale across distributed systems, and remain vulnerable to side-channel attacks. Their rigid design and development complexity further limit flexibility and broader adoption.

2.3 Related Work

MPC Data Analytics: Multiparty Computation offers a way to perform analytics without compromising data privacy. [47] demonstrated how MPC can be used for secure statistical analysis, such as regression analysis, by distributing the computation among multiple parties to ensure data privacy. [60] introduced a framework for privacy preserving deep learning using MPC, which allows multiple parties to collaboratively train a neural network without revealing their datasets. [45] proposed SecureML, a system that enables privacy-preserving machine learning inference using MPC. Their work demonstrates that performing secure prediction with minimal overhead is feasible. [64] presented a federated learning approach that leverages MPC to ensure the confidentiality of local model updates during the aggregation process. [44] is a mixed protocol framework designed to enable efficient and privacy-preserving machine learning, combining arithmetic and Boolean secret sharing techniques. [14] introduced techniques to reduce the communication complexity of MPC protocols, making them more practical for large-scale applications.

TEE Data Analytics: Trusted Execution Environments provide a secure area within a processor that ensures the confidentiality and integrity of data and code. [10] explored the use of Intel SGX for secure data aggregation, enabling analytics on encrypted data without exposing raw data to the aggregator. [48] demonstrated using TEE to perform privacy-preserving machine learning, allowing models to be trained on sensitive data without compromising privacy. [25] presented techniques for processing encrypted video data within TEE without exposing the content to the host system.

Video Analysis: Several deep learning architectures have been developed for object detection, video classification, and action recognition in videos. YOLOv3 [51] uses a robust backbone network based on Darknet-53 for real-time object detection, while Faster R-CNN [53] significantly improves the speed of object detection. 3D Convolutional Networks (3D CNNs) [63] process video frames as a 3D stacked frame volume, including temporal information for video analysis. Non-local Neural Networks [72] capture long-range dependencies in images and videos. Visor [49] provides confidentiality for both the user's video stream and the machine learning models in the event of a compromised cloud platform. Vigil [80] is a real-time distributed wireless surveillance system that leverages edge computing capabilities. Chameleon [28] is a controller that dynamically selects optimal configurations for existing neural network-based video analytics pipelines. EdgeEye [42] is an edge computing framework tailored explicitly for real-time intelligent video analytics applications. Reducto [36] is a system that adaptively adjusts filtering decisions based on time-varying correlations in the video data. CrossRoI [22] utilizes the inherent physical correlations of cross-camera viewing fields to enhance video analytics performance. Spatula [27] enables scalable cross-camera analytics by leveraging edge compute boxes. Lastly, PERQS [29] addresses the challenges of managing and querying CCTV footage in a contributory network.

No extensive studies have compared MPC frameworks and TEE frameworks specifically for shared video analytics workloads. Although both MPC and TEE offer robust methods for ensuring data privacy and security, their comparative effectiveness and performance in video analytics remain underexplored.

3 Overview and Threat Model

Collaborative video analysis is becoming increasingly important with the growing use of CCTV cameras. However, sharing data between different parties like government agencies and private organizations raises serious privacy concerns. Video files are large, making secure sharing and joint processing more demanding than other types of data.

On top of that, the complex machine learning models needed for video analysis make it harder to build systems that are both private and scalable. While prior work such as PERQS [29], Privid [7], RTFace [70], and VISOR [49] has explored secure video analysis, these systems often focus on local or isolated data processing. Our work builds upon PERQS [29], which uses local machine learning inference on CCTV streams to avoid raw video sharing but lacks support for joint and cross-organizational video analytics. To bridge this gap, we explore using MPC and TEE as foundational technologies for enabling privacy-preserving collaborative video analytics, offering strong security guarantees.

Our key contribution is identifying the most suitable implementation strategies for both approaches. In the case of MPC, a wide array of protocol variants exists. While prior research has explored general MPC performance characteristics, their suitability for collaborative video analytics tasks, particularly at the edge, remains underexplored. This work systematically examines the MPC design space to identify the most practical and efficient protocols for such workloads. In contrast, TEE-based approaches follow a more uniform design paradigm, where secure enclaves-whether through AMD-SEV, Intel-SGX, AWS-Nitro, or ARM TrustZone-process sensitive data in a hardware-isolated environment, ensuring confidentiality and integrity. Although conceptually similar, these implementations differ in their architectural design, affecting performance, integration complexity, and resilience to side-channel threats. To enable a robust evaluation, we design five real-world case studies that capture diverse challenges in joint video analytics, allowing us to analyze both MPC and TEE solutions regarding performance, security guarantees, and implementation complexity.

3.1 Case Studies

We selected five case studies to capture a diverse range of challenges in joint video analytics, each representing distinct computational requirements and privacy constraints. The primary goal is to enable collaborative analytics without exposing raw video feeds, which often contain sensitive information. Table 1 summarizes the critical computations, implementations, and available datasets. Critical computations are offloaded to either the MPC or TEE frameworks, while other application parts are executed locally on edge devices. To ensure a fair evaluation, we maintain consistent implementation across both approaches. More details on each case study are discussed below.

① Vehicle Re-identification (Re-ID) [43, 35, 67, 78]: Vehicle re-identification aims to recognize the same vehicle across multiple cameras in a network, playing a vital role in traffic management, law enforcement, and stolen vehicle recovery. Previous approaches have used various features, such as vehicle color, make, model, and license plate patterns, for matching. We use a feature extractor that utilizes deep learning models to generate feature vectors locally at the edge devices. These feature vectors are then compared against other video feeds to identify matches. We employ the Cosine Similarity function to measure the similarity between feature vectors. Since vehicle features contain sensitive information, the joint execution of cosine similarity must be conducted securely using MPC or TEE to preserve privacy.

② Scene Similarity Detection [81]: Scene similarity detection is crucial in video analytics to determine whether different cameras capture the same or similar environments.

Table 1. Details of the case studies include their critical computations, which are to be evaluated under privacy-preserving conditions, along with the available implementations and datasets. We use CityFlowV2 [16] for our analysis.

Case Study		Description	Critical Computation	Implementations	Datasets
①	Vehicle Re-Identification	identifying the same vehicle in another video camera.	Cosine Similarity Vector	DyeNet [35], PROVID [43], Siam R-CNN [67]	DAVIS dataset [50]
②	Scene Similarity	Detect weather two views are the same scene, or same events.	Euclidean Distance Measure	Factor Analysis [73], PDH Similarity [24],	Video Dataset [73]
③	Joint Recognition	Recognise a car which is partially visible in two video streams.	Video Stitching and Recognition	Object Centered Stitching [23]	Stitching Dataset [79]
④	Scene 3D Reconstruction	Merge different views from multiple cameras into a 3D scene and run query on it.	Scene 3D Reconstruction and query	EVolT [69], MR Video Fusion [82], DyNeRF [34]	Plenoptic Datasets [34], 3D Shapenets [74]
⑤	Count the Vehicles	From the overlapping video feeds find the number of vehicles in a certain interval.	Multi-video Object Counting	Argus [76], Rt3c [71]	CityFlow V2 [16], Cro HD dataset [62]

Since we do not trust any participants unquestioningly, the scene similarity check on overlapping scenes can reach a collective root of trust. Prior works [73, 24] explored various feature extraction methods for scene comparison. In our approach, we use a video summarizer to extract visual features as a vector, and then we use Euclidean distance to measure the distance between two scenes. We use the distance measure to compare and find the most similar past scenario for critical event detection. Scene features contain sensitive information and cannot be shared directly, so distance computation must be performed in a privacy-preserving manner.

③ Joint Object Recognition [51]: This case study focuses on recognizing objects, such as cars, bicycles, or trucks, which are spread across multiple cameras. Collaborative object recognition can enhance accuracy by combining video feeds from different sources securely [23]. For example, two cameras might capture complementary views of an object, improving performance. The videos are combined, and then object recognition is on top of the combined view. Here, the video stitching and recognition parts are the critical components that must be executed securely to protect video privacy.

④ Scene 3D Reconstruction [57, 82, 34, 69]: Scene reconstruction involves piecing together video data from multiple cameras to create a comprehensive 3D view of a specific area or event. For instance, in the case of a traffic accident, video data from different angles can be collaboratively analyzed to reconstruct the sequence of events. For the experiment, we used a static angle, rule-based 3D reconstruction, and then performed queries on this reconstructed 3D scene. Since the input videos are private, 3D reconstruction and querying must be done securely.

⑤ Counting the Total Number of Vehicles [76, 71]: Vehicle counting is crucial in traffic management, urban planning, and intelligent transportation systems. Traditional methods rely on individual cameras counting vehicles independently, which may lead to duplicate counts when the exact vehicle appears in multiple camera feeds. In this case study, we detect and count vehicles locally at each camera and securely remove duplicates by comparing vehicle features from different camera feeds. Feature extraction is performed on each detected vehicle to generate a feature vector representing its characteristics. To eliminate duplicate counts, we employ Cosine Similarity to compare feature vectors across multiple camera detections. We use MPC or TEE to ensure that feature comparisons and duplicate removals are performed without exposing sensitive data, enabling privacy-preserving collaborative vehicle counting.

3.2 Threat Model

- **MPC Threat Model:** For evaluating MPC defines the following threat models: **Semi-Honest Model:** Where participants adhere to the protocol specifications but may attempt to learn additional information from received messages or intermediate computations. The system assumes that these participants do not alter their behavior maliciously. **Malicious Model:** Where participants may deviate from the protocol, intentionally sending incorrect messages or manipulating computations to compromise the integrity or confidentiality of the data. In both models, there can be an honest or dishonest majority. In an honest majority setting, the majority of participants are assumed to behave honestly during any protocol execution. In the context of our work, remote servers, if involved, are treated as semi-honest, meaning they follow the protocol but may try to infer private inputs.
- **TEE Threat Model** To evaluate TEE, we assume that secure hardware provided by third-party servers (either owned by participants or external entities) is considered trustworthy and free of back-doors or vulnerabilities. Participants encrypt their private inputs before sending them to the TEE-protected enclaves for processing. Computations are securely performed within the enclave; only the final results are shared with the participants. Potential attacks, such as side-channel attacks on TEE, network eavesdropping, or breaches of secure enclaves, are treated as out of scope.

Both threat models assume that communication channels are secure and that cryptographic measures are in place to protect data transmission. Physical access to devices or servers hosting the computation is assumed to be restricted. Any compromise of the underlying hardware or cryptographic primitives is beyond the scope of this evaluation.

4 Experiment Setup

This section outlines the experimental setup and benchmark details for evaluating privacy preserving video analytics methods. We aim to identify the most suitable options for secure video analytics at edge devices by benchmarking video processing workloads. The MPC protocols were evaluated across different configurations.

Setup for MPC: Secure Multiparty Computation (MPC) configurations are evaluated on several key aspects. Depending on the security model, MPC can operate under a semi-honest or malicious setting. The number of participants also influences configuration choices, ranging from 2-party MPC, where two entities interact, to multiparty

Table 2. MPC protocols evaluated in this project. We use the MP-SPDZ [31] implementation.

Protocol	GC /SS	Adversary Type	Honest Majority	Domain	#	Details
yao [75, 77]	GC	Semi-honest	No	Binary	2	half-gate garbling
real-bmr [33]	GC	Malicious	No	Binary	2+	MASCOT protocol using BMR
semi-bmr [5]	GC	Semi-honest	No	Binary	2+	semi-honest version of real-bmr
rep-bmr [5, 33]	GC	Semi-honest	Yes	Binary	3	replicated sharing using BMR
semi2k [38]	SS	Semi-honest	No	Mod 2^k	2	OT-based Beaver triples
semi-bin [38]	SS	Semi-honest	No	Binary	2	bit-wise multiplication triples
semi [38]	SS	Semi-honest	No	Mod Prime	2	OT-based prime Beaver triples
mama [32]	SS	Malicious	No	Mod Prime	2	MASCOT with several MACs
mascot [32]	SS	Malicious	No	Mod Prime	2	OT correlation checks & MACs
spdz2k [15]	SS	Malicious	No	Mod 2^k	2	more efficient than MASCOT
ring [39]	SS	Semi-honest	Yes	Mod 2^k	3	replicated ring shares
rep-field [39]	SS	Semi-honest	Yes	Mod Prime	3	replicated field shares
atlas [21]	SS	Semi-honest	Yes	Mod Prime	3+	ATLAS sharing
shamir [12, 9]	SS	Semi-honest	Yes	Mod Prime	3+	Shamir secret sharing
rep4-ring [39]	SS	Semi-honest	Yes	Mod 2^k	4	replicated sharing

# - Number of participants supported, SS - Secret Sharing, GC - Garbled Circuit

setups, such as 3-party MPC, which leverage an honest majority for efficiency. The details of the protocols we considered in our evaluations are in Table 2. The configuration options we considered in our evaluations are as follows:

① **Adversary Type (Active/Passive):** MPC protocols are available against semi-honest adversaries or malicious adversaries. Similarly, protocols that assume an honest majority leverage optimizations to enhance performance. To evaluate these claims, we explored different protocols that provide security against semi-honest and malicious adversaries, considering scenarios where an honest majority is present or not.

② **Computation Domain (Arithmetic/Boolean):** Arithmetic-based MPC protocols are generally more efficient for integer addition and multiplication tasks. In contrast, Boolean circuits can be better when computations involve logical operations. Given that our workloads consist of numerous integer and matrix operations, we conducted evaluations using image processing macro benchmarks to compare the performance of arithmetic-based and Boolean-based protocols.

③ **Computation Choice (Secret Sharing/Garbled Circuit):** Secret-sharing based computations require multiple rounds of communication, with complexity increasing depending on the operation. In contrast, garbled circuits execute computations in a single round but incur significantly higher costs for arithmetic operations due to their reliance on binary representation. Our evaluations aim to determine whether garbled circuits or secret sharing is superior to our workloads.

④ **Number of Parties:** We evaluate the performance of the 2-party, 3-party, and 4-party protocols. Communication complexity generally increases as the number of participants increases. However, honest majority assumptions cannot be applied in a 2-party setting, while 3-party and higher setups can exploit these assumptions for performance optimizations. Since it remains unclear which n-party protocols will perform best for image processing workloads, we conduct extensive evaluations to determine the most suitable configuration.

⑤ **In-house/Outsourced model:** In an in-house MPC model computations take place entirely within local edge devices. Alternatively, an outsourced MPC model delegates computations to external servers. Our evaluation considers the trade-offs between these two approaches in the context of video analytics.

Setup for TEE: We consider a cloud-based TEE architecture, as illustrated in Figure 2. In this setup, edge participants offload their computations to a secure enclave hosted on a remote server. Before initiating any computation, participating organizations verify the authenticity and integrity of the TEE hardware through remote attestation [10]. Once verified, they establish a secure, end-to-end encrypted connection directly to the enclave, ensuring that private data inputs are transmitted securely. This design prevents a compromised host operating system from accessing or extracting sensitive data. After collecting all video or feature inputs from the involved parties, the enclave executes the joint computation. Finally, the results are securely transmitted to the edge participants while maintaining data confidentiality.

Benchmarks: To find out the most suitable MPC protocol, we use five macro benchmarks for image processing [55]. These benchmarks-thresholding, histogram, sobel edge detection, convolution, and thinning-represent fundamental operations commonly used in image and video processing. Each benchmark captures a distinct aspect of image analysis, such as feature extraction, edge detection, and transformation, making them well suited to assess the computational and communication overhead of different MPC protocols. We aim to identify the most reliable and efficient MPC protocol for edge devices by analyzing their performance.

Then for the remaining part of the TEE vs. MPC experiments, we focus on privacy-sensitive video analytics case studies, as explained in Table 1. We utilize the PERQS [29] framework to evaluate case studies, extending its query language to support joint video analytics. When a joint analysis query is executed, our MPC or TEE implementation is used to process the query securely. To compare the relative performance of both approaches across all five case studies, we measure execution time and global communication overhead for key computations involving private data. Beyond performance metrics, we also evaluate the security guarantees that each method provides and their practical implications. Additionally, we analyze the implementation complexity required to integrate MPC and TEE protocols into these applications securely.

5 Evaluation

This section outlines the implementation setup and the experimental evaluation. We begin by benchmarking various MPC protocols with image processing workloads. Followed by evaluating five case studies.

Implementation Details: We used the MP-SPDZ [31] framework for the implementation of MPC protocols, and Gramine-SGX [65, 66] and Linux-SGX [26] for Intel-SGX based TEE. All experiments were carried out on an Intel(R) Core(TM) i7-7700 CPU (3.60GHz) with four cores and two threads per core (8 hyper-threads), an 8192KB cache, and 32GB of RAM. Fedora 40 (Linux 6.13) was used for the MPC experiments, while Ubuntu 20.04 LTS (Linux 5.8) was used for the Gramine-SGX/Linux-SGX im-

Table 3. MPC protocols performance varying sharing scheme and adversary type. We have measured execution time and global data sent for all protocols. Some results are unavailable because the MP-SPDZ [31] does not support it. The highlighted values are the best time and global data communication for 2-party and 3-party protocols.

Applications	Secret Sharing Based				Garbled Circuit		Malicious Adversary	
	semi2k	semi-bin	ring*	rep-field*	yao	rep-bmr*	mascot	spdz2k
Matrix Operations	**0.03 s**	0.18 s	0.03 s	**0.02 s**	0.11 s	1.78 s	1.64 s	1.49 s
	0.5 mb	4.3 mb	**0.02 mb**	0.26 mb	9.12 mb	520 mb	352.9 mb	277.4 mb
Thresholding	**1.36 s**	2.61 s	**0.08 s**	0.43 s	1.75 s	39.12 s	569.3 s	232.7 s
	202.9 mb	**46.5 mb**	**8.1 mb**	26.2 mb	83.9 mb	4.90 gb	104.9 gb	40.8 gb
Histogram	**1.93 s**	3.08 s	**0.07 s**	0.456 s	2.12 s	46.8 s	488.4 s	181.7 s
	278.8 mb	**103.5 mb**	**9.8 mb**	24.81 mb	268.3 mb	4.25 gb	94 gb	30.3 gb
Sobel	**27.2 s**	NA	**4.23 s**	4.36 s	NA	NA	568.3 s	450.8 s
	7.23 gb	NA	**28.8 mb**	57.67 mb	NA	NA	135 gb	118.1 gb
3x3 Convolution	**13.9 s**	NA	4.43 s	**4.12 s**	NA	NA	354.3 s	274.1 s
	3.56 gb	NA	**14.5 mb**	28.92 mb	NA	NA	66.5 gb	58.1 gb
Thinning	**4.96 s**	7.85 s	**0.11 s**	0.58 s	16.8 s	198.6 s	610.6 s	278.9 s
	1.25 gb	967.8 mb	**22.1 mb**	54.17 mb	2.34 gb	326.4 gb	124.3 gb	58.2 gb

* 3-party protocols

plementation. We also have an ARM-based AWS-Nitro-enabled c6g.large Linux instance with AWS-Amazon 2023 (Linux 6.1) for Nitro-based TEE evaluation.

For the evaluation of MPC protocols, we used five image processing macro operations [55]: thresholding, histogram, Sobel edge detection, convolution, and thinning, along with a simple matrix multiplication workload. These benchmarks represent core computational tasks commonly used in image and video processing. For the experimental validation of our five case studies, we used the AICITY21 benchmark dataset (CityFlowV2) [16, 8], which features real-world traffic surveillance data collected from 46 cameras. The dataset contains 880 annotated vehicles in six different scenarios, with 215.03 minutes of video footage. We focused on eight junctions where multiple cameras provide overlapping recordings, enabling comprehensive multi-camera analysis.

Choosing the Best MPC Protocol: We utilized six workloads for our evaluation, matrix operations (including simple matrix additions and multiplications) and five fundamental image processing functions. To evaluate the image processing functions, we used a 256×256 monochrome image divided into four equal 128×128 pieces, each assigned to a different participant. For the 2-party setting, each party received two pieces of the image, while in the 4-party setting, each participant held one piece of the image. Then, a joint image processing operation was performed across all pieces, ensuring collaborative analysis while maintaining privacy. The MPC protocols listed in Table 2 were evaluated for these computations, with detailed results provided in Table 3 and Table 4. We have measured the total execution time, including the online and offline phases and the total global data communication. The MPC evaluation concerning the different design configurations is as follows.

① **Adversary Type (Active/Passive):** Malicious-secure protocols require significantly more execution time and communication overhead. As the complexity of operations increases, the communication cost grows exponentially. For example, in Sobel edge de-

Table 4. MPC protocol performance when number of parties increases.

Applications	2-party		3-party			4-party		
	semi2k	semi	ring	atlas	shamir	rep4-ring	atlas	shamir
Matrix Operations	**0.03 s**	0.11 s	**0.03 s**	0.06 s	0.04 s	0.07 s	0.08 s	0.05 s
	0.5 mb	15.9 mb	**0.02 mb**	0.88 mb	0.56 mb	0.3 mb	1.95 mb	1.12 mb
Thresholding	1.36 s	2.46 s	**0.08 s**	6.03 s	3.10 s	0.15 s	6.65 s	3.65 s
	202.9 mb	258 mb	**8.08 mb**	105.1 mb	86.5 mb	20.3 mb	207.8 mb	170.5 mb
Histogram	1.93 s	2.17 s	**0.07 s**	4.81 s	3.32 s	0.18 s	5.64 s	4.03 s
	278.8 mb	194 mb	**9.8 mb**	89.7 mb	75 mb	23.7 mb	177 mb	147.4 mb
Sobel	27.2 s	94.4 s	**4.23 s**	14.9 s	7.19 s	6.52 s	24.5 s	9.72 s
	7.23 gb	21.5 gb	**28.8 mb**	76.6 mb	57.7 mb	57.7 mb	153.1 mb	115.3 mb
3x3 Convolution	13.9 s	53.5 s	**4.43 s**	13.81 s	6.62 s	6.18 s	23.1 s	8.84 s
	3.56 gb	10.6 gb	**14.5 mb**	38.4 mb	28.9 mb	28.9 mb	76.9 mb	57.8 mb
Thinning	4.96 s	16.1 s	**0.11 s**	6.14 s	3.56 s	0.34 s	7.75 s	4.15 s
	1.25 gb	3.39 gb	**22.1 mb**	141.3 mb	114.5 mb	47.2 mb	278.2 mb	224.3 mb

tection, all three malicious-secure protocols required over 100 GB of data exchange, making them impractical for complex video analytics workloads.

② **Computation Domain (Arithmetic/Boolean):** Arithmetic-based protocols consistently outperform Boolean-based protocols in image/video processing tasks, as they involve extensive matrix operations. Boolean protocols are more efficient for numerous comparisons (e.g., thresholding) operations. For thresholding, the *semi-bin* protocol exhibited a lower communication overhead than *semi2k*, but for all other workloads, *semi2k* outperformed *semi-bin*. Additionally, MP-SPDZ currently does not support direct matrix multiplication for binary-shared inputs, preventing the execution of two workloads-Sobel and Convolution.

③ **Computation Choice (Secret Sharing/Garbled Circuit):** Secret Sharing based protocols consistently achieve lower execution time and reduced communication overhead across all workloads. As the complexity of integer operations increases, secret sharing is significantly more efficient than garbled circuits, which struggle with arithmetic intensive computations.

④ **Number of Parties:** The 3-party replicated secret-sharing protocol (*ring*) achieved the fastest execution time and lowest communication overhead across all workloads. The 2-party setting lacks optimizations available in 3-party honest-majority settings, making it less efficient for our use case. As the number of parties increases, both execution time and global data communication also increase.

⑤ **In-house/Outsourced model:** In our protocol, edge devices participate in computations but have limited resources and infrastructure. An outsourced model, as illustrated in Figure 1, shifts the computation to cloud servers. Here, edge devices only share input secret shares, significantly reducing their communication and computation load. This is particularly advantageous for DVR recorders and smart cameras, which lack the processing power required for joint video analytics using MPC.

Based on our evaluations, the Secret-Sharing-based 3-party replicated protocol (***ring***) is the most suitable choice for privacy-preserving video analytics workloads. We will use this protocol for our comparative analysis against TEE-based approaches.

Table 5. Performance of joint video analytics tasks of MPC and TEE. We used videos from CityFlowV2 [8] for all five workloads. Because of the lack of support and inherent memory limitations we are unable to run ml models using Intel SGX.

Case Study		TEE (Intel SGX)		TEE (AWS-Nitro)		MPC (MP-SPDZ)	
①	Vehicle Re-Identification	0.002 s	98.2 kb	0.001 s	98.2 kb	0.006 s	218.2 kb
②	Scene Similarity	0.003 s	123.4 kb	0.0012 s	123.4 kb	0.015 s	0.36 mb
③	Joint Recognition	NA	NA	4.1 s	37.3 mb	347.53 s	5.08 gb
④	Scene 3D Reconstruction	NA	NA	7.3 s	34.3 mb	621.4 s	8.56 gb
⑤	Count the Vehicles	0.025 s	1.3 mb	0.017 s	1.3 mb	0.033 s	2.1 mb

MPC vs TEE Performance Evaluation: In Case Study 1 (Vehicle Re-identification), vehicle features are extracted locally at the edge devices, followed by a privacy preserving computation of cosine similarity to identify matching vehicles across multiple cameras. For Case Study 2 (Scene Similarity), scene-level video features are extracted at the edge, and the Euclidean distance between these features is computed securely. Case Study 3 (Collaborative Object Recognition) combines two video streams using a predefined, rule-based stitching approach to generate a composite video, which is then processed using a vehicle recognition model. In Case Study 4 (3D Scene Reconstruction and Query), an ML model takes two input video streams to reconstruct a 3D scene representation. A subsequent query is then performed to detect specific objects within the reconstructed 3D environment. In Case Study 5 (Vehicle count), vehicles are first detected locally, and their features are extracted; the secure computation is then used to identify and eliminate duplicate detections across multiple camera feeds, ensuring an accurate count. For this evaluation, we used videos from the CityFlowV2 dataset [8], trimming them to similar lengths to ensure uniformity. We then averaged the execution time and communication cost across different sets of inputs.

The experimental results, as shown in Table 5, highlight the contrasting performance characteristics of MPC, and TEE-based implementations. For relatively lightweight workloads-such as vehicle re-identification (Case Study 1), scene similarity detection (Case Study 2), and total vehicle count (Case Study 5)-MPC protocols performed reasonably well. On average, they were 3 to 6 times slower than their TEE counterparts and incurred about twice the communication overhead. We extract features locally for these three case studies, so the computational cost of secure execution is relatively contained, making MPC a viable option despite its performance penalties.

In contrast, Case Studies 3 and 4, which require collaborative execution of deep learning models for object recognition and video scene stitching, posed significant challenges for MPC frameworks. These workloads demand intensive matrix operations, activation functions, and large intermediate state sharing, which are expensive in MPC due to multiple communication rounds and complex arithmetic over shared values.

As a result, MPC implementations were observed to be 70 to 90 times slower than TEE-based solutions. Moreover, the communication overhead ballooned into gigabytes, making the approach infeasible for real-time or bandwidth-constrained environments. However, TEE-based solutions efficiently handled these complex ML workloads, with only a few megabytes of private video input securely transferred to the enclave. Once inside, the entire model execution occurred at near-native speeds, providing a significant performance advantage. These results underscore the practical limitations of MPC in high-computation scenarios and demonstrate the strengths of TEE in handling complex, resource-intensive tasks.

Security Evaluation: MPC provides a robust and flexible security model that operates entirely in software and does not rely on trusted hardware components. This allows it to be deployed across a wide range of environments without dependence on specific hardware vendors or platforms. In addition, MPC protocols can be tailored to different threat models. The semi-honest model assumes that participants follow the protocol, and more secure configurations, such as protecting against malicious adversaries, can be adopted when stronger guarantees are required. These configurations introduce additional cryptographic checks and redundancy mechanisms, which increase computational and communication overhead but provide significantly enhanced security assurances.

In contrast, TEE such as Intel SGX or AWS Nitro Enclaves rely on secure hardware components to protect computations and data. TEE creates isolated enclaves where sensitive data can be processed securely, offering strong protection against software-level attacks and even against a compromised host operating system. However, their security is tied to the correctness and integrity of the hardware and its firmware, which introduces a dependency on hardware vendors. Establishing a root of trust with the manufacturer is necessary, and any vulnerabilities in the TEE implementation, such as side-channel attacks, speculative execution flaws (e.g., Spectre/Meltdown), or leakage through power analysis, can undermine the security guarantees. These limitations have led to concerns about TEE resilience.

Implementation Challenges: The MP-SPDZ framework offers a comprehensive and well-documented suite of tools to implement various MPC protocols. Its modular architecture and support for both arithmetic and Boolean computation domains make it suitable for rapid prototyping and evaluation. However, implementing machine learning (ML) workloads within this framework presents additional challenges. Specifically, developers must explicitly define the model architecture, layers, and activation functions compatible with the MPC back-end, which can be time-consuming and error-prone.

On the TEE side, Intel SGX poses several practical limitations. Most notably, SGX does not support system calls within the enclave environment, severely restricting the use of standard libraries and pre-existing machine learning frameworks such as PyTorch or TensorFlow. This limitation complicates the implementation of complex workloads that rely on dynamic memory management or file I/O. Furthermore, Intel has officially discontinued support for SGX on consumer platforms, reducing its viability for long-term and large-scale deployments. SGX enclaves also suffer from a memory limit (128MB of EPC memory), significantly hindering the execution of modern ML models that typically require larger memory footprints. Due to these limitations, we could not implement Python-based ML inference inside Intel SGX for our evaluation.

In contrast, AWS Nitro Enclaves provide a more flexible and developer-friendly environment. Nitro uses a container-based approach, allowing users to deploy secure enclaves by packaging their applications into Docker images. Which simplifies the enclave deployment process, especially for complex applications that depend on a large software stack. Additionally, Nitro Enclaves benefits from the scalability of the underlying EC2 infrastructure, users can allocate more CPU or memory resources to the enclave by selecting appropriate instance types. This makes it feasible to run memory-intensive ML workloads securely without modifying the application logic extensively.

6 Conclusion

We comprehensively evaluated Secure Multiparty Computation (MPC) and Trusted Execution Environments (TEE) as privacy-preserving solutions for joint video analytics in this work. We evaluated both approaches using five real-world case studies from traffic surveillance scenarios in the CityFlowV2 dataset. The case studies encompass a range of tasks, from simple similarity computations to complex multi-camera machine learning-based analytics, allowing us to evaluate the strengths and limitations of both security paradigms systematically.

Our experimental findings show that TEE consistently outperforms MPC implementations for workloads involving machine learning inference. This performance gap is particularly notable in case studies involving deep learning models. TEE-based implementations achieved up to 70-90× lower latency and required only a fraction of the communication overhead compared to MPC. TEE benefits from a centralized and hardware-isolated execution model, which enables efficient data handling and reduced interaction during computation. However, TEE is not without limitations. They require trusted hardware support and rely on vendor-specific root-of-trust mechanisms, which may not be feasible in all deployment environments. Additionally, their susceptibility to side-channel attacks and ongoing hardware vulnerabilities raises concerns about their suitability for highly sensitive applications. In contrast, MPC offers a software-only solution that can be deployed in environments where trusted hardware is unavailable or undesirable. Among the MPC protocols evaluated, secret-sharing-based 3-party protocols - particularly those exploiting honest majority assumptions - demonstrated the best performance in terms of both execution time and communication cost. While MPC remains less efficient than TEE for ML-based video analytics, it provides stronger fault isolation and more transparent trust assumptions.

In conclusion, while TEE is currently the preferred choice for compute-intensive video analytics tasks due to its performance benefits, MPC offers a viable and secure alternative for resource-constrained or hardware-agnostic settings. Selecting between the two approaches depends on the application environment's specific workload, deployment constraints, and trust model.

Acknowledgments

Grants from the Department of Science and Technology's National Mission on Inter-Disciplinary Cyber-Physical Systems (via IHUB-IIT Kanpur), the National Security Council (via Indian Urban Data Exchange), and the Indian Institute of Science funded our work.

References

1. Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: A survey. IEEE Internet of Things Journal **5**(1), 450–465 (2017)
2. Amazon Web Services: Aws nitro enclaves. https://docs.aws.amazon.com/enclaves/latest/user/nitro-enclave.html (2020), accessed: 2025-04-11
3. Arm: Arm platform security architecture. Arm Limited (2019)
4. Barr, J.: Introducing aws nitro enclaves. https://aws.amazon.com/blogs/aws/introducing-aws-nitro-enclaves/ (2020), accessed: 2025-04-11
5. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols. In: Proceedings of the 22nd annual ACM symposium on Theory of computing. pp. 503–513 (1990)
6. Bellare, M., Hoang, V.T., Rogaway, P.: Foundations of garbled circuits. In: Proceedings of the 2012 ACM conference on Computer and communications security. pp. 784–796 (2012)
7. Cangialosi, F., Agarwal, N., Arun, V., Narayana, S., Sarwate, A., Netravali, R.: Privid: Practical, Privacy-Preserving video analytics queries. In: USENIX Symposium on Networked Systems Design and Implementation (Apr 2022)
8. CHALLENGE, A.C.: Ai city challenge dataset 2021 (August 2021), https://www.aicitychallenge.org/2021-data-and-evaluation/ (accessed on 10 March 2023)
9. Chaum, D., Crépeau, C., Damgard, I.: Multiparty unconditionally secure protocols. In: Proceedings of the 20th annual ACM symposium on Theory of computing. pp. 11–19 (1988)
10. Costan, V., Devadas, S.: Intel sgx explained. Cryptology ePrint Archive (2016)
11. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 316–334. Springer (2000)
12. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 316–334. Springer (2000)
13. Cramer, R., Damgård, I.B., et al.: Secure multiparty computation. Cambridge University Press (2015)
14. Cramer, R., Damgård, I.B., et al.: Secure multiparty computation. Cambridge University Press (2015)
15. Cramer, R., Damgård, I., Escudero, D., Scholl, P., Xing, C.: SPDZ2k: Efficient MPC mod 2^k for dishonest majority (2018)
16. Fernandez, M., Moral, P., Garcia-Martin, A., Martinez, J.M.: Vehicle re-identification based on ensembling deep learning features including a synthetic training dataset, orientation and background features, and camera verification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2021)
17. Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, August, 2010. pp. 465–482. Springer (2010)
18. Gia, T.N., Jiang, M., Rahmani, A.M., Westerlund, T., Liljeberg, P., Tenhunen, H.: Fog computing in healthcare internet of things: A case study on ecg feature extraction. In: 2015 IEEE international conference on computer and information technology. IEEE (2015)
19. Goldreich, O.: Secure multi-party computation. Manuscript. Preliminary version **78**(110), 1–108 (1998)
20. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. STOC '85 (1985)
21. Goyal, V., Li, H., Ostrovsky, R., Polychroniadou, A., Song, Y.: Atlas: efficient and scalable mpc in the honest majority setting. In: Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021. Springer (2021)

22. Guo, H., Yao, S., Yang, Z., Zhou, Q., Nahrstedt, K.: Crossroi: Cross-camera region of interest optimization for efficient real time video analytics at scale. In: 12th ACM Multimedia Systems Conference. MMSys '21, Association for Computing Machinery (2021)
23. Herrmann, C., Wang, C., Bowen, R.S., Keyder, E., Zabih, R.: Object-centered image stitching. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
24. Hoi, C.H., Wang, W., Lyu, M.R.: A novel scheme for video similarity detection. In: Image and Video Retrieval: Second International Conference, CIVR 2003 Urbana-Champaign, IL, USA, July 24–25, 2003 Proceedings 2. pp. 373–382. Springer (2003)
25. Hunt, T., Song, C., Shokri, R., Shmatikov, V., Witchel, E.: Chiron: Privacy-preserving machine learning as a service. arXiv preprint arXiv:1803.05961 (2018)
26. Intel: Linux-sgx (2024), https://github.com/intel/linux-sgx
27. Jain, S., Zhang, X., Zhou, Y., Ananthanarayanan, G., Jiang, J., Shu, Y., Bahl, P., Gonzalez, J.: Spatula: Efficient cross-camera video analytics on large camera networks. In: 2020 IEEE/ACM Symposium on Edge Computing (SEC). pp. 110–124 (2020)
28. Jiang, J., Ananthanarayanan, G., Bodik, P., Sen, S., Stoica, I.: Chameleon: Scalable adaptation of video analytics. In: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. p. 253–266. SIGCOMM '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3230543.3230574
29. Joseph, A., Yadav, N., Ganapathy, V., Behl, D.: A contributory public-event recording and querying system. In: Proceedings of the Eighth ACM/IEEE Symposium on Edge Computing. p. 185–198. SEC '23, Association for Computing Machinery, New York, USA (2024)
30. Katz, J., Lindell, Y.: Introduction to modern cryptography: principles and protocols. Chapman and hall/CRC (2007)
31. Keller, M.: MP-SPDZ: A versatile framework for multi-party computation. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (2020)
32. Keller, M., Orsini, E., Scholl, P.: Mascot: faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 830–842 (2016)
33. Keller, M., Yanai, A.: Efficient maliciously secure multiparty computation for ram. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 91–124. Springer (2018)
34. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5521–5531 (2022)
35. Li, X., Loy, C.C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: Proceedings of the European conference on computer vision (ECCV). pp. 90–105 (2018)
36. Li, Y., Padmanabhan, A., Zhao, P., Wang, Y., Xu, G.H., Netravali, R.: Reducto: On-camera filtering for resource-efficient real-time video analytics. In: SIGCOMM '20. ACM (2020)
37. Lindell, Y.: Secure multiparty computation for privacy preserving data mining. In: Encyclopedia of Data Warehousing and Mining, pp. 1005–1009. IGI global (2005)
38. Lindell, Y.: Secure multiparty computation. Communications of the ACM **64**(1) (2020)
39. Lindell, Y., Nof, A.: A framework for constructing fast mpc over arithmetic circuits with malicious adversaries and an honest-majority. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 259–276 (2017)
40. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Annual international cryptology conference. pp. 36–54. Springer (2000)
41. Lindell, Y., Pinkas, B.: An efficient protocol for secure two-party computation in the presence of malicious adversaries. In: Advances in Cryptology-EUROCRYPT 2007: 26th An-

nual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007. Proceedings 26. pp. 52–78. Springer (2007)
42. Liu, P., Qi, B., Banerjee, S.: Edgeeye: An edge service framework for real-time intelligent video analytics. In: EdgeSys'18. Association for Computing Machinery (2018)
43. Liu, X., Liu, W., Mei, T., Ma, H.: A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, October 2016, Proceedings. Springer (2016)
44. Mohassel, P., Rindal, P.: Aby3: A mixed protocol framework for machine learning. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 35–52 (2018)
45. Mohassel, P., Zhang, Y.: Secureml: A system for scalable privacy-preserving machine learning. In: 2017 IEEE symposium on security and privacy (SP). pp. 19–38. IEEE (2017)
46. Muñoz, A., Rios, R., Román, R., López, J.: A survey on the (in) security of trusted execution environments. Computers & Security **129**, 103180 (2023)
47. Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., Taft, N.: Privacy-preserving ridge regression on hundreds of millions of records. In: 2013 IEEE symposium on security and privacy. pp. 334–348. IEEE (2013)
48. Ohrimenko, O., Schuster, F., Fournet, C., Mehta, A., Nowozin, S., Vaswani, K., Costa, M.: Oblivious {Multi-Party} machine learning on trusted processors. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 619–636 (2016)
49. Poddar, R., Ananthanarayanan, G., Setty, S., Volos, S., Popa, R.A.: Visor: Privacy-Preserving video analytics as a cloud service. In: 29th USENIX Security Symposium (USENIX Security 20). pp. 1039–1056. USENIX Association (Aug 2020)
50. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)
51. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. ArXiv **abs/1804.02767** (2018)
52. Ren, J., Zhang, Y., Zhang, K., Shen, X.: Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions. IEEE Communications Magazine **53**(3), 98–105 (2015)
53. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 1137–1149 (2015)
54. Roman, R., Lopez, J., Mambo, M.: Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems **78**, 680–698 (2018)
55. Sandler, M., Hayat, L., Costa, L.: Benchmarking processors for image processing. Microprocessors and Microsystems (1990)
56. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
57. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06). vol. 1, pp. 519–528. IEEE (2006)
58. Sev-Snp, A.: Strengthening vm isolation with integrity protection and more. White Paper, January **53**, 1450–1465 (2020)
59. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: Vision and challenges. IEEE internet of things journal **3**(5), 637–646 (2016)
60. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1310–1321 (2015)
61. Sumrall, N., Novoa, M.: Trusted computing group (tcg) and the tpm 1.2 specification. In: Intel Developer Forum. vol. 32. Intel Santa Clara, CA, USA (2003)
62. Sundararaman, R., De Almeida Braga, C., Marchand, E., Pettre, J.: Tracking pedestrian heads in dense crowd. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3865–3875 (2021)

63. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). Los Alamitos, CA, USA (dec 2015)
64. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., Zhou, Y.: A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM workshop on artificial intelligence and security. pp. 1–11 (2019)
65. Tsai, C.C., Porter, D.E., Vij, M.: {Graphene-SGX}: A practical library {OS} for unmodified applications on {SGX}. In: 2017 USENIX Annual Technical Conference (2017)
66. Tsai, C.C., Porter, D.E., Vij, M.: Gramine-sgx (2024), https://github.com/gramineproject/gramine
67. Voigtlaender, P., Luiten, J., Torr, P.H., Leibe, B.: Siam r-cnn: Visual tracking by re-detection. In: IEEE/CVF conference on computer vision and pattern recognition (2020)
68. Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., Vasilakos, A.V.: Software-defined industrial internet of things in the context of industry 4.0. IEEE Sensors Journal (2016)
69. Wang, D., Cui, X., Chen, X., Zou, Z., Shi, T., Salcudean, S., Wang, Z.J., Ward, R.: Multi-view 3d reconstruction with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5722–5731 (2021)
70. Wang, J., Amos, B., Das, A., Pillai, P., Sadeh, N., Satyanarayanan, M.: A scalable and privacy-aware iot service for live video analytics. In: MMSys'17. Association for Computing Machinery (2017)
71. Wang, R., Hao, Y., Miao, Y., Hu, L., Chen, M.: Rt3c: Real-time crowd counting in multi-scene video streams via cloud-edge-device collaboration. IEEE Transactions on Services Computing (2024)
72. Wang, X., Girshick, R.B., Gupta, A.K., He, K.: Non-local neural networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017)
73. Watanabe, Y., Hochin, T., Nomiya, H.: Method of similarity retrieval of color videos based on impressions. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). pp. 1–6. IEEE (2016)
74. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
75. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). pp. 160–164 (1982)
76. Yi, J., Acer, U.G., Kawsar, F., Min, C.: Argus: Enabling cross-camera collaboration for video analytics on distributed smart cameras. IEEE Transactions on Mobile Computing (2024)
77. Zahur, S., Rosulek, M., Evans, D.: Two halves make a whole: Reducing data transfer in garbled circuits using half gates. In: EUROCRYPT 2015. Springer (2015)
78. Zapletal, D., Herout, A.: Vehicle re-identification for automatic video traffic surveillance. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 25–31 (2016)
79. Zhang, F., Liu, F.: Parallax-tolerant image stitching. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3262–3269 (2014)
80. Zhang, T., Chowdhery, A., Bahl, P.V., Jamieson, K., Banerjee, S.: The design and implementation of a wireless video surveillance system. In: MobiCom '15. Association for Computing Machinery (2015)
81. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. Advances in neural information processing systems (2014)
82. Zhou, Y., Cao, M., You, J., Meng, M., Wang, Y., Zhou, Z.: Mr video fusion: interactive 3d modeling and stitching on wide-baseline videos. In: Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology. pp. 1–11 (2018)