IEEE ComSoc IEEE Open Journal of the Communications Society

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version 11 January, 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.011100

A Survey on Privacy and Security in Distributed Cloud Computing: Exploring Federated Learning and Beyond

Ahmad Rahdari [1] *(Member, IEEE)*, Elham Keshavarz[2], Ehsan Nowroozi[3] *(Senior Member, IEEE)*, Rahim Taheri[4] *(Senior Member, IEEE)*, Mehrdad Hajizadeh[5] , Mohammadreza Mohammadi[6], Sima Sinaei[6] AND Thomas Bauschert[5]

[1]School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
[2]Department of Computer Engineering, Pishtazan Institute of Higher Education, Shiraz, Iran
[3]Centre for Sustainable Cyber Security (CS2), University of Greenwich, United Kingdom
[4]PAIDS Reserach Centre, School of Computing, University of Portsmouth, United Kingdom
[5]Chair of Communication Networks, Technische Universität Chemnitz, Chemnitz, Germany
[6]RISE Research Institutes of Sweden

CORRESPONDING AUTHOR: Mehrdad Hajizadeh (e-mail: mehrdad.hajizadeh@etit.tu-chemnitz.de)

This work was supported by SUSTAINET-Advance (16KIS2280) and 6G-RIC (16KISK032) projects.

ABSTRACT The increasing need to process large, high-dimensional datasets and the substantial computational power required have made the use of distributed cloud servers essential. These servers provide cost-effective solutions that make storage and computing accessible to ordinary users. However, they might face significant vulnerabilities, including data leakage, metadata spoofing, insecure programming interfaces, malicious insiders, and denial of service. To gain public trust in distributed computing, addressing concerns related to privacy and security while ensuring high performance and efficiency is crucial. Multiparty computation, differential privacy, trusted execution environments, and federated learning are the four major approaches developed to address these issues. This survey paper reviews and compares these four approaches based on a structured framework, by highlighting recent top-tier research papers published in prestigious journals and conferences. Particular attention is given to progress in federated learning, which trains a model across multiple devices without sharing the actual data, keeping data private and secure. The survey also highlights federated learning techniques, including secure federated learning, by detecting malicious updates and privacy-preserving federated learning via data encryption, data perturbation, and anonymization, as new paradigms for building responsible computing systems. Finally, the survey discusses future research directions for connecting academic innovations with real-world industrial applications.

INDEX TERMS Distributed Cloud Computing; Edge Computing; Privacy-Preserving Computing; Federated Learning; Multi-Party Computation; Differential Privacy; Trusted Execution Environments.

I. INTRODUCTION

IN recent years, many service providers and companies have adopted cloud-based services to perform tasks, store data, and manage online information. These services typically rely on centralized setups with computing power, storage, and networks in big data centers. However, as connected devices grow rapidly through the Internet of Things (IoT), emerging new applications, such as self-driving cars, video surveillance, Augmented Reality (AR), and Virtual Reality (VR), have highlighted the limitations of this centralized cloud model [1]. For instance, AR and VR require very fast response times, often in milli- or nanoseconds, which remote cloud services struggle to meet owing to latency. In addition, transmitting large volumes of IoT data to the cloud can overload the network bandwidth. Consequently, there is a growing need to move beyond traditional cloud models to create faster, more efficient, and cost-effective solutions. New computing concepts, such as edge computing, cloudlets, and fog computing have been introduced to address these challenges [2]. These approaches reduce bandwidth conges-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

TABLE 1. Abbreviations and Definitions

Abbreviation	Definition	Abbreviation	Definition
AI	Artificial Intelligence	LSTM	Long Short-Term Memory
AQs	Analytical Questions	MitM	Man in the Middle
AR	Augmented Reality	MPC	Multi-Party Computation
BGW	Ben-Or, Goldwasser, Wigderson	OMTP	Open Mobile Terminal Platform
CSPs	Cloud Service Providers	PaaS	Platform-as-a-Service
DCC	Distributed Cloud Computing	PoPs	Points of Presence
DDoS	Distributed Denial of Service	PPC	Privacy-Preserving Computing
DoS	Denial of Service	RANs	Radio Access Networks
DP	Differential Privacy	ROM	Read Only Memory
FL	Federated Learning	SaaS	Software-as-a-Service
GAN	Generative Adversarial Network	SDN	Software-Defined Networking
GPUs	Graphics Processing Units	SFE	Secure Function Evaluation
IaaS	Infrastructure-as-a-Service	SSS	Shamir's Secret Sharing
IAI	Industrial Artificial Intelligence	TAs	Trusted Applications
IDS	Intrusion Detection System	TEE	Trusted Execution Environments
IID	Independent and Identically Distributed	TOS	Trusted Operating System
IIoT	Industrial Internet of Things	VIM	Virtual Infrastructure Manager
IoT	Internet of Things	VR	Virtual Reality
IPS	Intrusion Prevention System	WoS	Web of Science

tion and enhance latency by handling data closer to its source rather than relying on distant cloud data centers. While the distinctions between these paradigms are still debated within the research community, they all contribute to a broader shift towards decentralizing cloud computing, known as Distributed Cloud Computing (DCC) [3]. The development of 5G networks has made this trend by enabling the processing of larger data volumes more efficiently, particularly through innovative architectures such as distributed Radio Access Networks (RANs) and Software-Defined Networking (SDN) [4]. A distributed RAN architecture breaks down traditional RAN functions into smaller, distributed units. This allows for flexible deployment, improved network performance, and reduced latency. Likewise, SDN complements this by decoupling the control plane from the data plane, enabling dynamic resource allocation and enhanced network programmability. Today, Microsoft Azure, Google Cloud, IBM Cloud, VMware Cloud, Oracle Cloud, and Amazon Web Services offer affordable and scalable cloud solutions that provide private and public access to data via the Internet [5].

Although DCC systems offer numerous advantages, they also intensify the privacy and security challenges of centralized cloud systems and may introduce new attack vectors. In DCC, raw data from different decentralized nodes are often sent to a central server for processing. Taking healthcare as an example, sensitive patient data from various hospitals can be shared with a central cloud to train machine learning models. This process can increase the risk of data breaches, unauthorized access, or interception during transmission and storage because the raw data passes through multiple points where it can be exposed. Furthermore, DCC systems manage massive datasets spread over different servers. This raises serious security concerns, as the presence of a few malicious or colluding workers could jeopardize all data and put it at risk [6]. Users often provide sensitive information, which they expect to be protected. Simultaneously, they might want to share more data to make their experiences smarter and better. For example, in AR, providing a system with more data can lead to better object recognition, personalized content, and a more realistic experience [7]. However, as more data are added, it becomes more difficult to protect the data. This makes it important to prevent unauthorized access to and misuse of data during collection, processing, and analysis. The question is: "Can DCC benefits be achieved while addressing privacy and security?"

This survey study presents Federated Learning (FL) as a promising solution to these privacy and security issues. FL is a fast-growing distributed learning model that runs across multiple data sources and addresses privacy concerns by keeping raw data locally stored with the participants. The main idea behind FL is that model hyperparameters, such as gradients and loss functions, can be shared and protected more easily, while still providing the necessary information for improving a global model. In simple terms, FL allows different participants to collaborate through the exchange of model updates, rather than sharing raw data during training [8]. By combining FL with DCC, we can enhance data processing and bring about significant advancements in data science. DCC is required to address the limitations of centralized clouds by bringing resources closer to users and improving latency, scalability, and compliance, especially for data-intensive applications. FL complements DCC by enabling collaborative Artificial Intelligence (AI)

model training without centralizing sensitive data, preserving privacy, or meeting regulatory requirements. Together, DCC and FL provide scalable, efficient, and secure solutions for modern decentralized computing needs. Because the distributed nature of DCC systems intrinsically aligns with FL, it has gained significant popularity in academic research and industry.

FL should be considered as part of a wider area of Privacy-Preserving Computing (PPC) techniques. In computer science, PPC includes various technologies that allow for computation while keeping participants' privacy safe. Since the 1970s, researchers have been working on PPC, focusing on finding a balance among security, performance, accuracy, and efficiency [9]. The four main categories from different angles or generations of PPC include Multi-Party Computation (MPC), Differential Privacy (DP), Trusted Execution Environments (TEE), and FL. The classification of PPC into different generations in the literature is based on broad paradigms that represent fundamental architectural shifts in how privacy is protected. Each generation reflects a major evolution in privacy-preserving methodologies, addressing the limitations of its predecessors, and introducing a new way of thinking about privacy protection. These generations are often combined. For example, MPC, DP, and TEE can work with FL to create techniques, such as FL-MPC, FL-DP, and FL-TEE [17]. In this survey, however, we examine these generations individually, as each offers a distinct perspective on the problem and has its strengths and weaknesses. By comparing FL with other privacy-preserving generations, this study highlights its advantages, limitations, and potential as a key solution for secure distributed cloud environments. We also introduce other FL techniques, such as secure FL, by detecting malicious updates and privacy-preserving FL via data encryption, data perturbation, and anonymization, as new paradigms for building responsible computing systems. Our goal is to present a thorough evaluation by highlighting recent top-tier research papers published in various prestigious journals and conferences and offering valuable suggestions for future work and real-world applications. In doing so, we hope to bridge the gap in understanding the intersection of FL and DCC and provide insights into building responsible PPC systems.

A. ANALYTICAL QUESTIONS

This study includes the following Analytical Questions (AQs) and provides clear and straightforward answers for each:

AQ1: What are the key advantages of FL over MPC, DP, and TEE when addressing privacy and security challenges in DCC?

AQ2: What challenges arise when using FL in DCC, and how can they be mitigated?

AQ3: What are the advantages and disadvantages of each secure FL technique, including methods for detecting malicious updates, data encryption, data perturbation, and anonymization?

AQ4: How can FL establish a fair balance between security, performance, efficiency, and accuracy in DCC?

B. CONTRIBUTIONS

Figure 1 illustrates the graphical abstract for privacy-preserving and secure DCC, which will be discussed in this survey. The primary contributions of this survey are outlined below:

- We present a comprehensive review of the cutting-edge privacy and security challenges and proposed solutions in DCC, including an introduction to different PPC generations and an analysis of underlying philosophies guiding their integration.
- While other studies often focus on a single PPC technique, such as MPC, DP, or TEE, our survey covers all of these alongside FL, offering a comprehensive assessment of the available strategies.
- We analyze recent top-tier research papers from prestigious conferences and journals on MPC, DP, TEE, and FL to provide a detailed overview of current privacy and security techniques in DCC.
- We introduce a structured evaluation framework that compares the methods proposed in recent top-tier research papers across six key dimensions: privacy, security, scalability, maturity, advantages, and limitations, providing an objective assessment of their practical deployment considerations.
- We highlight FL as a key method for mitigating security risks in DCC systems and position it as a more suitable solution than other methods in this domain.
- We present a taxonomic structure of privacy-preserving and secure FL techniques, including methods such as malicious update detection, data encryption, data perturbation, and anonymization, and outline their main strengths and weaknesses in real-world DCC applications.
- We discuss open research directions in the privacy and security of DCC to bridge the gap between academic advancement and industry requirements.

Section II reviews existing surveys that focus on privacy and security in DCC and FL. It also highlights how our study differs from earlier studies. Section III provides a basic overview of DCC and their structure. Section IV examines the vulnerabilities associated with the DCC systems. Section V explains the methods used to search for topics, including relevant keywords, and how we chose the bibliometric database. Section VI analyzes the four generations of PPC, featuring key research papers from leading conferences and journals, and evaluating their strengths and weaknesses using our structured framework. Section VII discusses the AQs proposed in the Introduction. Section VIII suggests directions for future research and exploration. Finally, Section IX

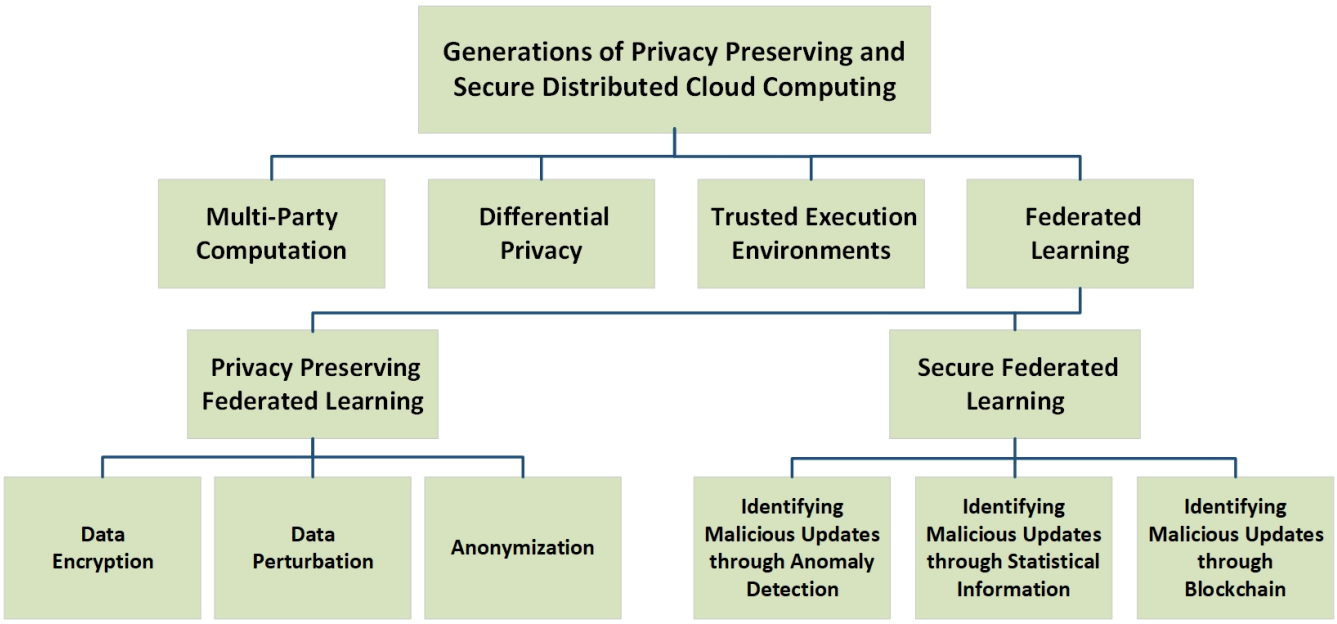


FIGURE 1. **Graphical abstract for privacy-preserving and secure DCC.**

concludes the study by summarizing the main insights and contributions.

II. EXISTING SURVEYS

This section presents a collection of relevant and related studies that review the use of cloud computing or federated learning in the security field. Next, we compare our work with existing surveys.

In [9], the authors evaluated the foundations, objectives, details, architectures, and implementation of privacy-preserving FL for computing. However, they did not provide a comprehensive view of DCC. In [10], the authors conducted a methodical review of privacy-preserving FL and examined potential privacy breaches, but did not thoroughly explore its integration with DCC and compared it with other techniques such as MPC, DP, or TEE. In [11], the applications of FL for edge computing were reviewed considering existing research problems and their possible solutions. This article lacks depth in comparing FL with other critical PPC approaches such as MPC and DP. In [12], the authors analyzed the routing protocol, architecture, and hardware requirements of FL in edge computing, demonstrating its practicality through case studies, but failing to provide a comprehensive comparison of all PPC techniques. In [13], the authors provided an overview of FL methods with a specific focus on edge devices and their computational constraints. They discussed various FL frameworks, challenges related to hardware heterogeneity, and communication issues. However, their study primarily focused on FL implementation and scalability without extensively addressing the security challenges and privacy-preserving techniques. In [14], the authors reviewed security and privacy issues in decentralized FL, emphasizing how blockchain-based architectures could mitigate server-related vulnerabilities. Although the survey effectively categorized security mechanisms, it did not provide a comparative analysis of FL with other PPC approaches, which limited its applicability to broader DCC paradigms. In [15], the authors reviewed PPC techniques in FL, categorizing different inference attacks and discussing countermeasures, such as adversarial training and homomorphic encryption. Although this work systematically analyzed privacy challenges, it did not thoroughly examine FL integration with cloud computing. In [16], the authors explored the integration of FL and edge computing in IoT applications, highlighting various cryptographic techniques, perturbation-based privacy mechanisms, and adversarial training approaches. However, this survey lacked an analysis of FL with other PPC techniques. The authors of [17] presented a survey on robust FL and categorized existing schemes without discussing solutions to privacy and security challenges across DCC. In [18], the authors investigated FL and addressed its challenges, privacy risks, and security solutions, such as secret sharing and quantum FL methods, with limited attention to the broader range of PPC techniques, narrowing its scope in a DCC context.

A. DIFFERENCES FROM EXISTING SURVEYS

Unlike previous surveys, our research emphasizes the privacy and security challenges related to DCC. We highlight the key solutions proposed by researchers to address these concerns. Although some studies have focused on traditional cloud computing, our study is the first to specifically investigate security and privacy techniques in DCC. Our review distinguishes itself from existing surveys by emphasizing its practical and real-world applications. Although many studies

TABLE 2. The differences between our work and existing surveys.

Ref.	Privacy	Security	FL	Cloud Computing	Description	Differences from Our Work
[9]	✓	✓	✓	×	Provided an overview of FL and its application in PPC.	This review did not provide a comprehensive view of DCC. In contrast, our review integrates FL with DCC for a broader perspective.
[10]	✓	×	✓	×	Surveyed privacy-preserving FL.	This review presented a study on FL but did not thoroughly explore its integration with DCC and comparison with other PPC techniques. In contrast, our review delves deeper into FL within the context of DCC.
[11]	✓	✓	✓	✓	Reviewed FL challenges and potential solutions in edge computing.	This article lacks depth in comparing FL with other PPC approaches. Our review, however, provides a more thorough exploration of FL in the context of DCC.
[12]	✓	✓	✓	✓	Surveyed FL implementations in edge computing.	This survey failed to provide a comprehensive comparison of all PPC techniques. Conversely, our review emphasizes the advantages of FL when considered as a solution for DCC challenges.
[13]	×	×	✓	✓	Provided an overview of FL with a focus on edge devices and their computational constraints.	This review does not deeply explore privacy-preserving computation techniques or the security challenges in DCC, which our work addresses comprehensively.
[14]	✓	✓	✓	×	Examined security and privacy issues in decentralized FL and discussed blockchain-based approaches.	This survey focused on security threats in decentralized FL but did not explore the comparison of FL with other PPC techniques, which our review covers in depth.
[15]	✓	✓	✓	✓	Surveyed various privacy-preserving computation protocols in FL.	This work focuses primarily on FL security threats and mitigation methods, whereas our review expands on FL's role in DCC and compares multiple PPC approaches.
[16]	✓	✓	✓	✓	Investigated security and privacy-preserving techniques in FL within edge IoT environments.	This paper discussed cryptographic solutions in FL but lacked a holistic comparison of FL with other PPC techniques, which our work provides.
[17]	✓	✓	✓	×	Surveyed the strengths and weaknesses of various privacy-preserving FL techniques.	This study concentrated on FL solutions using blockchain without discussing it across DCC. Our review explores FL within the broader context of DCC.
[18]	✓	×	✓	×	Reviewed privacy challenges and preservation solutions in FL.	This study did not consider cloud computing, which limited its scope. Our work, however, concentrates on privacy and security in DCC.
Our work	✓	✓	✓	✓	Reviews privacy and security vulnerabilities in DCC, proposed PPC solutions (i.e., MPC, DP, TEE, and FL), and their comparison, highlighting the advantages of FL and providing an overview of open research directions.	Not Applicable (NA).

have investigated FL, we consider it a key solution for security issues in DCC systems. Typically, other studies reviewed FL separately; however, our review includes all four PPC techniques: FL, MPC, DP, and TEE. Therefore, one of the key advantages of our review is the comprehensive comparison of the strengths and weaknesses of the four major PPC techniques. After comparing these methods in the context of DCC, we found that using FL in DCC is the most effective way to address security and privacy challenges. We also highlight FL techniques such as secure FL through the detection of malicious updates and privacy-preserving FL via data encryption, data perturbation, and anonymization. While previous studies have tended to focus on individual methods or specific applications such as IoT, our research integrates these techniques into a unified framework and shows how they can be combined or adapted to meet the unique challenges of DCC.

This comprehensive overview not only broadens the scope of privacy and security considerations but also provides practical insights for both researchers and industry professionals. Furthermore, none of the existing review articles has fully addressed the implementation challenges and practical uses of DCC. Here, we highlight real-world examples and outline the research directions required by the industry. Our study identifies gaps and provides a roadmap for future research, emphasizing the need for scalable, efficient, and secure solutions for evolving DCC environments. This focus makes our review particularly relevant to developers, engineers, and policymakers tasked with ensuring the security and privacy of the distributed systems. Table 2 compares our findings with those of other surveys in this area.

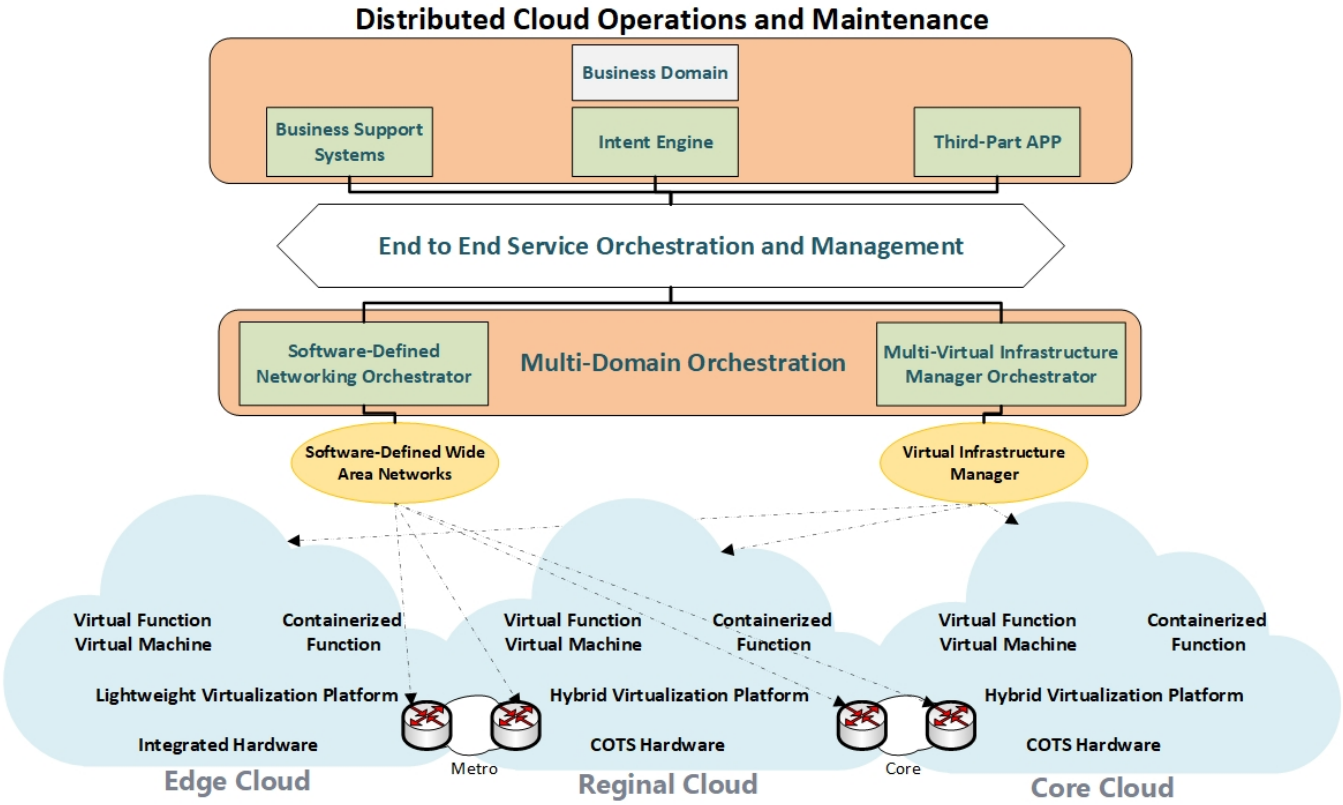


FIGURE 2. **Distributed cloud architecture.**

III. BACKGROUND ON DCC

A distributed cloud is a modern cloud computing model in which Cloud Service Providers (CSPs) augment their cloud services to encompass different physical locations beyond their usual data centers. It allows clients to operate their applications or parts of them in various places, such as public clouds, edge locations, and data centers, to meet specific needs such as faster replies and legal requirements. Even though the services are spread out, CSPs are still responsible for operating and maintaining the distributed cloud infrastructure. The widespread adoption of cloud technology is primarily owing to its scalability and affordability. The growth of DCC is now closely tied to the increasing use of big-data applications. Solutions powered by the DCC can provide valuable insights, aiding in more precise decision-making. DCC is also recognized for its manageability and scalability, offering independent and on-demand access to network resources and connectivity [19].

As shown in Figure 2, the architecture of a distributed cloud follows a three-layered network structure: core, regional, and edge clouds [1], [20]. The core cloud manages the overall coordination and control of the distributed cloud, thereby addressing heavy workloads and permanent storage. The regional layer, which sits between the core and edge layers, supports edge clouds in the same region, helps with traffic load balancing by caching data, and improves service quality. It can also host network services such as the 5G core, while the edge cloud, which operates at the network's edge, supports services such as IoT and RAN, which require low latency. CSPs such as Google, AWS, and Azure own the core cloud layer, which acts as the central point for managing cloud resources. These companies might also have some edge Points of Presence (PoPs), but they often partner with third-party telecom providers to expand their edge infrastructure. Regional clouds can be set up at CSP's PoPs or in partner data centers to serve specific regions, whereas edge clouds are deployed at the network edge to serve end users who need low-latency services. Each layer of a distributed cloud has a different infrastructure. The core and regional layers typically use standard commercial hardware and hybrid platforms with virtual machines and containers. However, edge clouds, which have more limited resources and operate in harsher environments, rely on lightweight virtualization platforms (e.g., Unikernels and MicroVMs) and time-sensitive networking for deterministic communication [21]. Additionally, edge clouds must address environmental factors such as power efficiency, thermal constraints, and intermittent connectivity, requiring advanced workload migration strategies and energy-aware scheduling policies. To meet the performance demands of specific applications, they may also integrate hardware accelerators such as SmartNICs for packet processing, TPUs for AI inference,

and FPGAs for workload offloading. These specialized accelerators enable edge computing environments to support high-performance applications, such as real-time video analytics and autonomous system control.

To ensure seamless coordination among cloud layers, distributed clouds require comprehensive orchestration mechanisms that manage computing, networking, and storage resources dynamically. This orchestration is critical for maintaining service-level agreements and adapting to changing network conditions. The orchestration framework consists of multiple layers, including a service orchestrator, workload scheduler, and cross-domain controller, each responsible for different aspects of cloud resource management. The service orchestrator abstracts infrastructure complexity by providing an end-to-end view of available resources, dynamically allocating tasks based on real-time demand. A multidomain orchestrator with cloud schedulers ensures efficient task execution by integrating different orchestration modules, including the Virtual Infrastructure Manager (VIM) orchestrator and the Software-Defined Networking (SDN) orchestrator. The VIM orchestrator oversees cloud-native platforms such as OpenStack, Docker, and Kubernetes, managing virtualized resources across geographically distributed cloud nodes. It dynamically provisions and scales workloads, ensuring optimal resource utilization and balancing workloads between cloud layers. The SDN orchestrator integrates software-defined wide-area networks that manage distributed SDN controllers across network domains, providing dynamic traffic engineering, latency-aware routing, and automated fault recovery. Unlike traditional networking approaches, SDN-enabled distributed clouds allow for programmable network control, improving adaptability to changing network conditions and enhancing security through policy-driven enforcement [22], [23]. Advanced traffic optimization strategies, such as intent-based networking further enhance network performance and resilience against disruptions. Together, these orchestrators provide a comprehensive view of resources and make it possible to deliver end-to-end services that combine cloud-based tasks (as data processing and storage) and network tasks (as data transmission).

The distributed cloud offers three distinct service delivery models: Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS), and Platform-as-a-Service (PaaS). Each of these models addresses various business needs and offers unique advantages, as outlined below:

- **SaaS:** Allows customers to utilize cloud-based software applications via the internet. With SaaS, setup expenses and fundamental infrastructure management costs are eliminated and updates are handled efficiently. However, customers possess minimal security because the technical infrastructure and implementation platform are externally managed. The SaaS model concentrates on policy-driven access management, in which users are typically only permitted to download specific information from applications. This approach makes the service available to a large number of users. Common examples of SaaS include Google Drive, Microsoft 365, Dropbox, and Zoom, which provide seamless access to essential business and collaboration tools through the cloud [24].
- **IaaS:** Involves the virtual provision of computing resources, including hardware, networking, and storage services. This model encompasses operating systems, operational services, and specific network components, all of which fall under customer management. Managed exclusively by a CSP, IaaS is notable for its application in security fields such as Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), firewalls, and virtual machine monitoring. A prime example of this is Amazon Web Services, which delivers these services through its cloud platform, allowing businesses to manage their software [25], [26].
- **PaaS:** Facilitates the development and deployment process, offering a more streamlined and convenient approach. Essentially functions as an online computing platform, allowing CSPs to manage the underlying infrastructure, including networks, storage, servers, and operating systems. Meanwhile, customers retain some control over the applications they deploy and potentially some configuration settings. When compared to SaaS and IaaS, PaaS provides enhanced extensibility and greater customer control over security. PaaS integrates various operating systems and application servers, such as Microsoft Azure, Google App Engine, and the LAMP stack (Linux, Apache, MySQL, PHP) [27], [28].

Distributed cloud also comes in distinct deployment models, generally categorized as public, private, hybrid, and community, each offering different levels of control, security, and customization.

- **Public cloud:** A third-party vendor hosts a public cloud application in a data center and offers resources over the Internet shared by organizations and individuals who wish to use or purchase them. Public cloud services provide varying levels of security and confidentiality that may not be suitable for handling highly sensitive data or specific use cases. Most offerings in IaaS, PaaS, and SaaS fall under the public cloud model. These services allow users to access computing power, applications, and complete platforms remotely through a web browser or terminal without the need to host or maintain software or hardware on-site. While some public cloud resources are free, others may require payments through subscription plans or on a pay-as-you-go basis. This type of cloud is a cost-effective option for users because of its multitenant nature, flexibility, scalability, and location independence [29], [30].
- **Private cloud:** Organizations use the private cloud model to meet the cloud needs of various departments. This approach allows them to maintain their infrastruc-

ture to deliver cloud services. The private cloud model can operate within the organization's network or by authenticating users at the firewall. These clouds provide significantly greater control, security, and customization compared to public options. The higher level of control simplifies compliance with regulations concerning sensitive data and offers better protection for trade secrets and internal communications. However, the nature of private clouds also means they tend to be more costly and labor-intensive to establish and maintain. Thus, this model is ideal for large enterprises and government agencies to secure critical data [31], [32].

- **Hybrid cloud:** This type combines the speed and security benefits of a private cloud with the cost-effective computing and storage advantages of a public cloud. In this setup, companies can store sensitive information and vital applications on a private cloud that meets the regulatory requirements. Simultaneously, they can use a public cloud for less sensitive tasks. In addition, a hybrid cloud enhances disaster recovery and backup options. However, in practice, managing a hybrid cloud can be difficult, especially when expanding the system. Every time a system updates or a new member joins the system, new challenges can arise [33].
- **Community cloud:** This type of cloud functions as a private cloud but is shared by multiple organizations, motivated by common operational or regulatory needs. Infrastructure can be owned, managed, and operated by one or more organizations within the community or by an external third party. Community clouds also offer advantages over public clouds, such as scalability and cost-effectiveness. A community cloud is a specialized type of hybrid cloud that serves the needs of a specific group. By concentrating on a particular purpose, community clouds can offer essential services while minimizing the complexities and challenges often associated with broader hybrid cloud models [34], [35].

Figure 3 compares the distributed cloud service and deployment models, highlighting trade-offs in security, access control, and cost. Public clouds offer affordability with limited security, private clouds provide security at a higher cost, hybrid clouds balance both, and community clouds serve specialized needs.

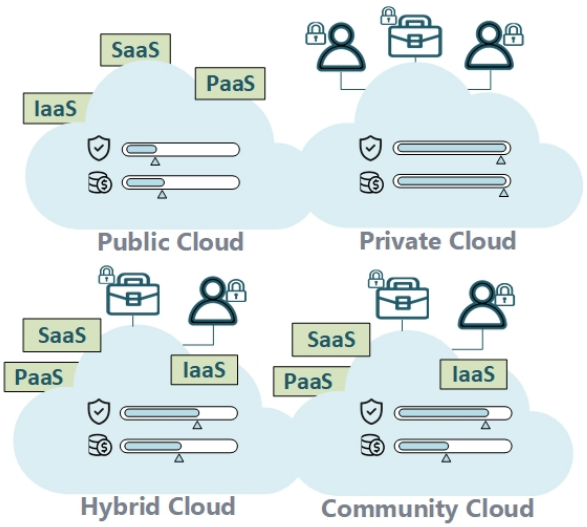


FIGURE 3. Overview of distributed cloud service delivery and deployment models, illustrating differences in security, access control, and cost using visual icons: shields for security, currency for cost, briefcases for tasks, user for restricted access, and locks for authentication mechanisms.

IV. DCC VULNERABILITIES

DCC, despite its many benefits, has several vulnerabilities that can jeopardize the privacy and integrity of data because of their reliance on multiple interconnected servers, one of which is the fragmentation of data across different geographical locations and cloud providers. This exposes the system to data breaches, especially during data transit between nodes, owing to insecure sharing methods and weak encryption [36]. To mitigate this, MPC can be employed to ensure that no single party ever has access to the complete dataset. Another approach is the application of DP, which introduces controlled noise into datasets, preventing adversaries from extracting sensitive information from aggregate outputs. However, the existence of multiple access points for data recovery makes it challenging to secure potential vulnerabilities in transit [37]. Furthermore, differences in security policies among cloud providers can cause misconfiguration and increase the risk of exposure. The presence of anonymous profiles also reduces control over regulatory compliance and auditing, leading to exposure to sensitive data [38].

Another significant vulnerability arises from the shared responsibility model of cloud services. While cloud providers are responsible for securing the infrastructure, users are accountable for securing their own data and applications. Many users either lack the expertise or awareness to implement adequate security measures. This often results in poor identity and access management [39]. One strategy to counteract this issue is the use of FL, which enhances security by ensuring that only model updates are shared. In addition, TEEs can be leveraged to execute code in isolated, hardware-protected environments. However, the complexity of managing distributed environments across multiple clouds also increases the likelihood of human error or mismanagement. Users are often unaware of risks such as phishing, which can result in unauthorized access to cloud systems [40].

Another critical area of vulnerability lies in programming interfaces, which, while facilitating user engagement in application development, also introduce weaknesses that unauthorized users can exploit. This complexity within the cloud framework often leads to backdoor access [41], [42]. Cloud providers should enforce strict gateway security policies, including rate limiting, authentication tokens, and access

control. Interface request anomaly detection, combined with AI-based behavioral analysis, can help identify and block malicious interactions in real-time. Service and account hijacking can occur when adversaries exploit vulnerabilities to access legitimate websites and reuse credentials for malicious purposes [43]. The potential for malicious insiders to manipulate or steal data also presents a significant risk because their high-level access can bypass many detection mechanisms. The issue of colluding workers also persists, as they may share confidential information or misuse their access to disrupt operations. To defend, Role-Based Access Control and Attribute-Based Access Control should be combined with continuous user behavior analytics. Secure audit trails with blockchain technology can further enhance the transparency and traceability of user actions. However, weak isolation mechanisms between workers, insufficient encryption techniques, and vulnerabilities in trust models can exacerbate this problem [44], [45]. In addition, services may be misused, especially in the PaaS and IaaS models, where control over user activity can be limited. If not properly managed, running multiple virtual machines on a single server in multi-tenant environments can also lead to unauthorized access [46].

DCC's distributed nature introduces further challenges, such as latency and coordination issues between nodes, which can be exploited to overwhelm systems with excessive traffic and disrupt services. This overload can cause significant financial and reputational harm to cloud service providers. Vulnerabilities related to network protection, such as poorly configured firewalls, often contribute to these security concerns [47], [48]. A mitigation strategy involves deploying AI-driven anomaly detection systems that continuously monitor traffic patterns.

The Other risks include side-channel vulnerabilities, where information may be leaked during process execution, and issues such as metadata manipulation, which can compromise the service confidentiality [49]. To counter this, techniques such as constant-time cryptographic operations can be enforced to ensure execution patterns do not reveal sensitive information. Hardware-based solutions, including TEEs, also play a crucial role in preventing attackers from gaining insights through cache timing or power consumption analysis. Furthermore, the use of noise injection methods in computations can obfuscate exploitable patterns. Metadata spoofing can be mitigated by implementing cryptographic signatures and authenticated encryption mechanisms. The deployment of blockchain-based integrity verification can further enhance the authenticity of metadata across distributed cloud networks. Moreover, reliance on third-party providers for cloud services introduces supply chain risks, where vulnerabilities in a provider's infrastructure can affect customer systems. providers' lack of transparency about their security practices adds another layer of uncertainty to customers [50]. A defense strategy against supply chain vulnerabilities involves enforcing contractual security requirements on third-party providers. Continuous monitoring of supplier security through Security Information and Event Management systems and independent audits can enhance supply chain resilience.

Overall, the roots of these vulnerabilities are insecure sharing methods, weak encryption, inadequate security measures, insufficient user awareness, and complexities in managing distributed systems. Thus, these systems have an inherently large surface area for potential security issues, making them susceptible to exploitation if not properly managed. Collectively, DCC vulnerabilities can be categorized based on the nature and part of the cloud infrastructure they affect, which are listed with an explanation and root cause analysis in Table 3.

V. METHODOLOGY

This section explains the research methodology used for this survey, including the relevant keywords, bibliometric databases, and evaluation framework.

A. SEARCHED KEYWORDS

To comprehensively explore the relevant field, we searched the literature using a range of keywords, including "Distributed Cloud Computing, "Edge Computing", "Fog Computing", "Cloudlets", "Privacy-Preserving Computing", "Multi-Party Computation", "Differential Privacy", "Trusted Execution Environments", "Federated Learning", "Collaborative Learning", "Distributed Machine Learning", "Secure Function Evaluation", "Multi-User Computation", "Data Perturbation", "Data Encryption", "Anonymization", "Additive Masking", "Secret sharing", "Homomorphic Encryption", "Malicious Updates", "Anomaly Detection", "Malicious Workers", and "Colluding Workers". We conducted this search using the Web of Science (WoS) and Scopus databases, focusing on articles from conferences and journals written in English. Furthermore, we ensured that all the references we included were from the last decade, from 2015 to the date of this survey's preparation.

B. SELECTION OF BIBLIOMETRIC DATABASE

Scopus and WoS were chosen as the key bibliometric sources for this study to ensure an extensive literature review. By incorporating both databases, we aimed to conduct a thorough exploration of academic publications, thereby improving the completeness and credibility of the survey. The review specifically focused on selecting research from well-regarded journals and conferences in the fields of DCC, FL, security and privacy, networking, information theory, and communication, including CCS, S&P, USENIX, ESORICS, ICML, NeurIPS, PMLR, ICCV/ECCV, NDSS, CVPR, ACM CCS, ACM CODASPY, ACM MobiCom, ITW, ISIT, INFOCOM, IEEE Cloud Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Mobile Computing, Computers & Security, IEEE Internet of Things

TABLE 3. DCC Vulnerabilities, their explanations and cause analysis.

Vulnerability	Explanation and root cause analysis
Data Leakage and Loss	DCC faces data leakage issues, leading to significant risks. Major causes of data loss include weak encryption and verification methods, damaged data centers, and poor disaster planning [51].
Anonymous Profiles	DCC reduces the need for direct hardware and software management, which can weaken security practices such as regulatory compliance, auditing, and system hardening. This lack of control increases the risk of sensitive data exposure, particularly when anonymous profiles are present [52].
Vulnerable Programming Interfaces	While these interfaces facilitate user engagement in application development, they also create vulnerabilities that unauthorized users can exploit through backdoor access. [53].
Services and Account Hijacking	An adversary sends a web service to an unauthorized website. This allows the adversary to access the legitimate site, reuse usernames and passwords, and conduct phishing [43].
Malicious Insiders	With high-level network access, these insiders can alter valuable and protected data. Intrusion detection systems and firewalls often fail to detect suspicious activity that appears legitimate [44], [54].
Colluding Workers	Workers may collaborate to breach data privacy, manipulate computations, or disrupt operations, resulting in unauthorized access and data leakage. This vulnerability arises from inadequate isolation between workers, weak trust models, and insufficient encryption to protect data during processing [45].
Abuse and Immoral Use	The DCC infrastructure offers storage and bandwidth; however, limited control can result in security weaknesses that unauthorized users may exploit. Because PaaS and IaaS involve a high level of user participation, these vulnerabilities have significant impacts [55].
Distributed Technology Vulnerabilities	Multi-tenant architecture enables multiple users to share a single application in a virtualized environment. However, adversaries may take control of authorized virtual machines, potentially disrupting the cloud's core operations and affecting its functionality [56].
Denial of Service (DoS)	Adversaries generate massive amounts of traffic to disrupt service accessibility. The main causes of this vulnerability include insufficient network protection, unmonitored traffic patterns, and weak firewall configurations [47].
Distributed Denial of Service (DDoS)	An advanced form of denial of service that overwhelms a target server with excessive traffic from multiple sources, leading to partial or total inaccessibility [48].
Probing	User information that can identify individuals may be at risk of data breaches due to insufficient monitoring, the multi-tenant nature of cloud environments, human error in configuration, and unsecured channels [57].
Man in the Middle (MitM)	An unauthorized entity may intercept the communication between users and cloud services. This vulnerability is often caused by weak encryption or poor authentication measures [58].
Phishing	Involves deceiving users into providing sensitive information, like login credentials, by directing them to malicious websites. This could result in unpermitted access to cloud systems, risking data and resources. Common causes include low user awareness, poor email security, and weak authentication protocols [59].
Side Channel	Targets the execution of computer processes and compromises data integrity by exploiting the side-channel information [49].
Zombie	Affects service availability by causing disruptions to legitimate virtual machines, either by directly flooding the host machine or through indirect methods [60].
Spoofing Meta Data	Undermines the confidentiality of services by altering the web service description, leading to irregular service behaviors [61].
Remote to Local Vulnerabilities	Occur when an unauthorized entity remotely gains local access owing to weak authentication and access controls, insecure protocols and configurations, unpatched software, and misconfigured cloud services [62].
User to Root Vulnerabilities	Occur when a user with limited access gains unauthorized root or administrative privileges owing to flaws in privilege mechanisms, misconfigured cloud services, unpatched software and security holes, weak access controls, and insecure configurations [63].

Journal, IET Information Security, ACM Transactions on Privacy and Security, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Big Data, IEEE Transactions on Services Computing, IEEE Transactions on Network Science and Engineering, Journal of Information Security and Applications, IEEE Internet Computing, Journal of Machine Learning Research, IEEE Journal on Selected Areas in Information Theory, IEEE Transactions on Information Theory, Neurocomputing, Journal of Parallel and Distributed Computing, Journal of Supercomputing, Future Generation Computer Systems, IEEE Transactions on Communications, IEEE Journal on Selected Areas in Communications, Computer Communications, IEEE/ACM Transactions on Networking, IEEE Transactions on Network and Service Management, IEEE Open Journal of the Communications Society, and other top-tier venues. This selection underscores our dedication to showcasing cutting-edge research from esteemed conferences and journals renowned for their significant contributions and strict standards in these advancing fields. To manage this literature methodically and comprehensively, we follow the following approach:

1) A keyword search will be conducted by combining terms related to the general concept "AND" terms specific to summarization papers. An illustrative example of such a search is as follows:

FIGURE 4. Methodology for searching topics within the taxonomy.

(TITLE-ABS-KEY((“Cloud Computing” OR “Edge Computing” OR “Privacy-Preserving Computing”) AND (“Federated Learning” OR “Multi-Party Computation” OR “Differential Privacy” OR “Trusted Execution Environments”)))

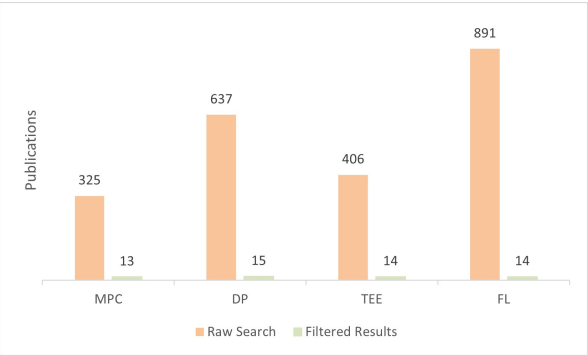


FIGURE 5. Publications that were found for MPC, DP, TEE, and FL using the methodology in WoS and Scopus over the past decade. ”Raw search” shows initial results, while ”Filtered results” indicate finalized counts after removing duplicates and filtering by quality and relevance to DCC.

2) Once the taxonomy is selected, each branch and subtopic is investigated using topic or subtopic names as keywords. A filtering approach similar to the previous method will be applied. Removing the keywords related to summarization publications will significantly increase the number of papers, so an additional filter based on the quality of venues and journals will be introduced to ensure that high standards are maintained. Figure 4 shows the applied process. After filtering, we reviewed all papers for final selection based on our expertise. The following example demonstrates the search for a topic by referencing a list of top-tier conferences and publications.

(TITLE-ABS-KEY((“Cloud Computing” OR “Edge Computing” OR “Privacy-Preserving Computing”) AND (“Federated Learning” OR “Multi-Party Computation” OR “Differential Privacy” OR “Trusted Execution Environments”)) AND (SRCTITLE(“IEEE Open Journal of the Communications Society” OR “IEEE Cloud Computing” OR “IEEE Transactions on Information Forensics and Security” OR “USENIX Security Symposium” OR “Neurocomputing” OR “ICML” OR “Journal of Supercomputing” OR “ACM Transactions on Privacy and Security” OR “ACM Conference on Data and Application Security and Privacy” OR “IEEE Transactions on Neural Networks and Learning Systems”))

C. THE EVALUATION FRAMEWORK

We introduce an evaluation framework to facilitate a rigorous comparison of PPC techniques in DCC. This framework enables a comprehensive and objective analysis of various MPC, DP, TEE, and FL methods proposed by the scholars. By assessing these methods through integrated metrics, we provide deeper insights into their practical deployment considerations, helping policymakers select the most appropriate mechanism based on their specific needs.

Our structured framework evaluates each study across six key dimensions: privacy, security, scalability, maturity, advantages (pros), and limitations (cons). **Privacy** examines whether the proposed approach in a study meets the necessary privacy criteria. Specifically, it considers data exposure—whether raw data is exposed at any stage or remains protected during computation. It also evaluates anonymization mechanisms, privacy-enhancing techniques, resistance to inference attacks, and the ability to prevent unauthorized data mining. **Security** assesses the resilience of each study against various vulnerabilities based on the reports provided. This includes the presence and strength of cryptographic algorithms, the robustness of security mechanisms such as encryption, and the use of secure protocols. It also considers the degree of reliance on external entities, such as centralized servers, cloud providers, or trusted intermediaries.

Scalability examines computational complexity by evaluating the processing power, memory, and bandwidth required to execute the proposed approach, as reported in the studies. It also assesses communication costs and their impact on

system performance. Furthermore, it considers deployment feasibility by focusing on how easily the technique can be integrated into distributed systems, cloud environments, or edge computing infrastructures. **Maturity** measures the stage of development of each method, categorizing them into three levels: *Concept* refers to theoretical proposals without real-world implementation. *Prototype* represents an early-stage implementation demonstrating feasibility in a controlled environment. *Experimental* techniques have been tested under real-world conditions, often accompanied by performance evaluations.

The **pros** metric highlights the key benefits of each technique, such as increased efficiency, reduced computational or communication costs, robustness under varying network conditions and adversarial settings, and compatibility with cloud computing services or resource-constrained edge devices. In contrast, the **cons** metric identifies key challenges and limitations, including high computational costs, significant processing and memory requirements, and scalability constraints.

VI. GENERATIONS OF PRIVACY-PRESERVING AND SECURE DCC

Privacy-preserving technologies have evolved over several generations, each building on the previous one to enhance data security and privacy in increasingly sophisticated ways. This section comprehensively overviews the key generations and most significant studies conducted in each area (chosen through the methodology). This highlights the strengths and weaknesses of each study.

A. MULTI-PARTY COMPUTATION (MPC)

The field of MPC, also called Secure Function Evaluation (SFE), began with Yao's millionaire problem [64], [65]. This famous problem asks how some millionaires can determine who has more wealth without revealing the exact amounts to each other. For instance, imagine three entities, Alice, Bob, and Charlie, each with a salary. They want to find out who earns the most without telling each other their actual salaries. Mathematically, this is similar to calculating the function $max(x, y, z)$, where x, y, and z are the salaries. If they had a trusted friend like Tony, they could tell him their salaries, and Tony would tell them the highest one. However, MPC allows Alice, Bob, and Charlie to determine the highest salary without Tony's involvement or disclosing individual salaries mainly based on Shamir's Secret Sharing (SSS) [66]. They only learn what they can infer from the results and inputs. This basic example can be expanded to situations in which participants have multiple inputs and outputs and where different parties receive distinct outputs from the function. In addition, the function being computed could be more complicated than simply determining the maximum [67]. Generally, in an MPC scenario, multiple entities, denoted as $e_1, e_2, \ldots, e_N$, possess secret data $(s_1, s_2, \ldots, s_N)$. Their objective is to collaboratively compute the outcome of a function $F(s_1, s_2, \ldots, s_N)$ while ensuring that their inputs remain confidential.

The two important goals of any MPC protocol are **input privacy** and **correctness**. Input privacy implies that no private data are revealed during the process, except for what can be gained from the outcome. Correctness means that even if some participants try to cheat or share information, they cannot make honest participants accept an incorrect result [68]. Some protocols guarantee correctness by ensuring that honest participants always obtain the correct result, whereas others allow participants to stop the process if they detect cheating. MPC protocols are often evaluated using the **real-world/ideal-world paradigm**. In the ideal world, a trustworthy party executes the result, whereas in the real world, participants exchange messages directly. A protocol is considered secure if real-world interactions reveal no more information than what would be revealed in an ideal world scenario [69].

MPC assumes that adversaries could be the participants themselves, who may try to collude and break the security. Let t represent the total number of parties in the protocol and m denote the number of potentially adversarial parties. The approaches and protocols for scenarios in which $m < \frac{t}{2}$ differ from those without this assumption [70], [71]. The latter includes significant cases, such as two-party computations in which one party might be compromised, as well as more general situations in which numerous participants might be corrupted and collaborate to undermine the honest parties. Different security models have been used to deal with other types of adversaries. For example, **covert security** assumes that adversaries will only try to cheat if they believe they will not be caught [72]. This model balances efficiency and security by ensuring that dishonest behavior is detected with high probability while still allowing for efficient computations. The security of MPC protocols relies on multiple factors, including the complexity of the computation, the type of network used, and message exchange. Adversaries can be *static*, targeting specific participants, or *dynamic*, where they change their targets during the process, making defending against them more difficult [73].

Despite its strong security guarantees, MPC presents significant computational trade-offs, particularly in terms of communication overhead and efficiency. Many protocols require extensive message exchange, leading to high bandwidth consumption. For instance, Garbled Circuits [74] involves transmitting large encrypted data, which can be impractical for bandwidth-limited settings. In contrast, SSS-based approaches reduce the computational complexity but require more communication rounds. Lighter methods, such as Oblivious Transfer [75], improve efficiency but rely on additional security assumptions. Another critical issue in MPC is scalability. Many protocols struggle to handle a growing number of participants because communication and computational costs increase significantly. Traditional thresh-

old schemes often exhibit $\mathcal{O}(n^2)$ complexity, which makes them inefficient for large-scale applications. To mitigate this problem, modern MPC frameworks employ precomputation and parallel processing. A real-world implementation of these optimizations is the Conclave Query System [76], an MPC-based secure data query system developed by the SAIL Lab at Boston University. Conclave mitigates MPC's inherent overhead by leveraging data parallelism, plaintext computation, and optimized secure MPC instructions. However, precomputation introduces challenges, such as storage overhead and vulnerabilities to adaptive adversaries who may exploit precomputed values [77].

Companies such as Unbound Tech, Cypherium, IBM, Intel, Kudelski Security, and R3 have recently founded an MPC Alliance to raise awareness and encourage the adoption of MPC technology. Significant studies in this domain have recently emerged. The authors of [78] explored the scenario of cooperative learning involving datasets protected with various keys, and presented a solution using MPC leveraging multi-key fully homomorphic encryption. Similarly, secure-multiparty privacy-computation-based collaborative learning was developed in [79] using the Diffie-Hellman key agreement and ElGamal encryption to provide both parameter and data privacy without compromising the efficiency of the output model. In [80], a privacy-preserving classification scheme that uses gated recurrent unit networks and MPC technology that relies on secret sharing was outlined. The authors of [81] developed an MPC framework to safeguard data privacy during the evaluation of complex polynomials on large matrices. They introduced a novel polynomial-sharing approach and demonstrated its efficacy in executing essential operations, such as matrix addition and multiplication. In [82], the authors integrated concepts from the Ben-Or, Goldwasser, and Wigderson (BGW) scheme with polynomial codes, leading to the creation of polynomial sharing. This innovative approach offers a secure method for computing arbitrary matrix polynomials while ensuring the confidentiality of the data matrices. In [83], a technique for secure data sharing was developed and designed to produce a single matrix result, while ensuring that the input matrices remain confidential and safeguarded against potential interception by adversaries. The authors of [84] introduced a multiparty neural network training framework that achieved linear communication complexity. This framework ensures end-to-end information-theoretic privacy through an iterative multiparty coded computing approach. In [85], the authors explored the range of optimal MPC costs by drawing a connection between covering codes and syndrome decoding. They offered an algebraic explanation, showing that the largest fraction of servers required to compute each subfunction became more constrained. In [86], the authors investigated distributed computations among several users based on linear separability, wherein N servers assist K users in calculating their desired functions. They established connections among matrix factorization, covering codes and syndrome decoding, thereby reducing computational and communication costs. The researchers in [87] introduced a secure way to share IoT data using MPC and blockchain smart contracts. Their approach helps prevent data leaks and unauthorized access in edge computing by ensuring only trustworthy IoT devices can participate by utilizing a Bloom filter. In [88], the authors presented a cloud-based system designed to protect privacy in distributed applications. It uses a Naïve Bayesian classifier along with multi-party random masking and polynomial aggregation to enhance security while avoiding the downsides of traditional encryption methods. The study in [89] focused on improving privacy-preserving authentication mechanisms for smart cities. Instead of relying on a single certification authority, their approach distributes identity management across multiple service providers using an MPC protocol and a pseudonym-based signature system. This setup prevents any single entity from controlling user credentials. Finally, Körner's characteristic graph method was applied in [90] to a promising multitask, multiserver distributed computing framework. The authors explored a scenario involving linearly separable functions and cyclic dataset placement and demonstrated significant performance improvements. According to the selected literature, Table 4 compares these MPC approaches using our structured evaluation framework.

B. DIFFERENTIAL PRIVACY (DP)

DP is a formalized framework designed to release statistical insights from datasets, while safeguarding individual privacy. It allows data controllers, such as CSPs, to disclose overall trends within a group without revealing specific information about individuals. This protection is achieved by introducing calibrated noise into statistical computations that ensure that the resulting statistics are still useful but prevent any meaningful inferences about individual data entries [91]. For example, an algorithm that computes various statistics (such as variance, median, and mean) on a dataset is considered differentially private if its output does not reveal whether it contains data from a specific individual.

DP was introduced by Cynthia Dwork and Frank McSherry in 2006, as outlined in their landmark studies [92] and [93]. Their work proposed a formal method for ensuring privacy in data analysis by highlighting the inherent tension between maintaining statistical accuracy and protecting privacy. As a result, one of the key computational trade-offs in DP is balancing privacy guarantees with data utility. Increasing the level of noise enhances privacy protection but simultaneously reduces the accuracy of analytical results. Moreover, answering too many queries, even random ones, about a database can inevitably lead to privacy breaches. The challenge is to formally define privacy to effectively understand and manage this trade-off.

Thus, DP is formally characterized by an accumulative risk model, rather than a binary one, meaning that every time a person's data are accessed or processed, the risk of exposing that individual increases slightly. This is where the

TABLE 4. Summary and Comparison of recent significant MPC approaches.

Ref	Privacy	Security	Scalability	Maturity	Description	Pros	Cons
[78]	✓	✓	×	Concept	A cloud computing scheme for privacy-preserving deep learning that employs double decryption and fully homomorphic encryption.	There is no need for data owners to be involved in decrypting the learning outcomes.	The computational and communication costs for the owner(s) are significant.
[79]	✓	✓	✓	Prototype	An approach for multi-party deep learning in cloud environments.	Protects both data and parameter privacy and is compatible with any deep neural network.	It is based on the assumption that gradients are provided by participants.
[80]	✓	✓	×	Prototype	Leverages MPC and gated recurrent units to facilitate relation classification.	Each cloud server takes part in the classification of relational data.	Tend to be inefficient because of their substantial computational demands. In addition, intricate service infrastructures may put users' privacy at risk.
[81]	✓	✓	✓	Concept	Utilizes a group of workers to divide the operation of calculating a polynomial function over large private matrices.	Ensuring the privacy of the data remains intact.	This scheme relies on randomness, which cannot be fully obtained or secured without the involvement of trusted entities.
[82]	✓	✓	×	Concept	Presents a new sharing method along with several operations, such as matrix multiplication, addition, and matrix transposition.	Delivers efficiency optimized for the order of fundamental operations like addition and multiplication.	Data accuracy and privacy are emphasized, but the computation complexity between workers is not minimized.
[83]	✓	✓	×	Concept	In this scheme, matrices are divided into blocks of varying sizes based on an entangled polynomial sharing protocol.	Allows fundamental operations such as addition, multiplication, and transposition to be executed privately.	The recovery threshold for polynomial codes does not increase the number of workers participating.
[84]	✓	✓	×	Prototype	A neural network training architecture for multiple parties that achieves linear complexity of communication and end-to-end privacy ensured by information theory.	Improves communication complexity from quadratic to linear and supports adversary tolerance and dropout resilience.	High communication complexity and iterative coded computing mechanisms are resource-intensive.
[85]	✓	✓	×	Concept	Investigates multi-user linearly-separable function computation with a relationship between covering codes and syndrome decoding.	Achieves a lower computation cost while preserving function accuracy.	Suffers from communication overhead in large-scale distributed systems.
[86]	✓	✓	×	Concept	Studies multi-user linearly-separable distributed computation and explores the relationship between computation cost and matrix factorization over finite fields.	Reduces computation and communication costs through sparse matrix factorization and coding-theoretic properties.	Limited scalability due to the complexity of sparse matrix operations.
[87]	✓	✓	✓	Prototype	A secure data-sharing approach by leveraging MPC and blockchain.	Provides a public verifiability mechanism and filters out non-trustworthy devices via Bloom filters.	Relies on a trusted third-party service.
[88]	✓	✓	×	Prototype	A cloud-based classification framework using multi-party random masking and polynomial aggregation.	Balancing privacy preservation and accuracy.	Still involves computational overhead in prior probability calculations.
[89]	✓	✓	×	Prototype	Develops an authentication mechanism for smart cities through MPC.	Eliminates the need for a trusted certification authority and ensures unlinkability between anonymous accounts.	Computationally expensive for large-scale implementations due to the complexity of the pseudonymization scheme.
[90]	✓	✓	×	Concept	Applies Körner's characteristic graph approach to multi-server multi-task distributed computation, showing gains in linearly separable functions and cyclic dataset placement.	Achieves considerable reductions in communication cost for multi-linear functions and cyclic dataset placements.	The gains are specific to certain types of functions.

parameters ϵ (epsilon) and δ (delta) are used to quantify the privacy loss or individual-level risk arising from the use of their data. Regardless of the additional information an adversary may possess, privacy is always bounded by these parameters [94].

Let ϵ represent a real number in the positive range and consider a randomized algorithm A that takes a dataset as input, representing the actions of the trusted party managing the data. The image of A, denoted by $im(A)$, refers to the possible outcomes of this algorithm. The algorithm is said to provide (ϵ, δ)-differential privacy if for any two

datasets D_1 and D_2 there is a difference of only one element (representing one person's data), and for every subset S of $\text{im}(A)$ [94], [95]:

$$\Pr[A(D_1) \in S] \leq e^{\epsilon} \Pr[A(D_2) \in S] + \delta \quad (1)$$

When $\delta = 0$, this is known as pure differential privacy, and the algorithm meets ϵ-differential privacy.

The concept of **composability** in DP allows the integration of multiple privacy-preserving mechanisms while maintaining the privacy guarantees of the overall system. In sequential composition, when a mechanism is queried multiple times, the cumulative privacy loss is the sum of the privacy parameters of individual mechanisms. For instance, if an ϵ-differential privacy mechanism is queried k times, the system ensures ϵk-differential privacy. By contrast, parallel composition applies to mechanisms working on distinct subsets of data, where the overall privacy loss is determined by the mechanism with the highest privacy cost [96], [99]. Here, a crucial property is **robustness to post-processing**, which ensures that once a mechanism satisfies a privacy guarantee, any further manipulation of its output, whether random or deterministic, does not reduce the privacy level. This maintains privacy guarantees throughout the data lifecycle even after subsequent transformations or analyses [100].

DP can be extended to protect groups of records through the concept of group privacy, which safeguards neighboring databases that vary over one record. As the number of differing records increases, the privacy loss increases. Specifically, if c records differ between the two datasets, then the privacy loss is bounded by $\exp(\epsilon c)$ rather than $\exp(\epsilon)$. By adjusting ϵ to ϵ/c in equation 1, the group as a whole receives ϵ-differential privacy protection, whereas each item within the group is protected by ϵ/c-differential privacy [101]. Because DP is probabilistic, it inherently relies on randomization. Mechanisms such as Laplace and exponential methods use controlled noise or sampling from problem-specific distributions to achieve privacy. **Sensitivity** is another key concept, which measures how much a function's output changes upon modification of a single dataset entry. For example, let f represent a mapping function linking datasets to real numbers and let Δf denote the sensitivity of f. Sensitivity is defined as [96]:

$$\Delta f = \max \|f(D_1) - f(D_2)\|_1 \quad (2)$$

Here, the maximum is computed across all the dataset pairs D_1 and D_2 which are distinct by a single element, and $\|\cdot\|_1$ represents the L_1-norm. In this context, if f computes a query on a medical database, for example, the sensitivity would be one, as changing any one data entry would alter the result by at most one. Sensitivity can be generalized to other metrics and is essential for designing differentially private algorithms, including those that use noise from Laplace or Gaussian distributions.

Achieving scalability in DP remains an issue, particularly when dealing with large-scale distributed data systems. The primary difficulty arises from the computational demands associated with generating private statistics for high-dimensional datasets. As the number of dimensions increases, the complexity of ensuring privacy increases exponentially, thereby requiring substantial computational power. To address this, several techniques have been developed. One such method is **moment accounting** [97], which provides a more precise estimation of cumulative privacy loss over multiple queries. Another approach is **privacy amplification via subsampling** [98], which leverages the fact that analyzing only a randomly selected subset of data can effectively reduce the overall privacy cost. However, these techniques require careful implementation and fine-tuning because improper configurations may lead to either excessive noise or insufficient privacy protection.

In recent years, significant research has been conducted on DP approaches for preserving the privacy of DCC applications. In [102], researchers developed an edge-driven framework for sensor cloud systems, emphasizing the differential processing of data on edge servers to protect privacy while reducing storage and communication costs. Their approach involves the DP of raw data from sensor networks and ensures that the core data remain protected, even if the cloud is compromised. In [103], a DP sustainable fog-oriented query model for computing data centers was proposed. Their model effectively balances privacy preservation and data utility. Their model demonstrated robust resistance to various privacy vulnerabilities by injecting Laplacian noise. In [104], the authors proposed a classification approach based on DP to manage sensitive information during data mining. Their experiments revealed enhanced iteration efficiency and highlighted the algorithm's reliability and timely response while maintaining strong privacy safeguards. In [105], the authors explored an Internet-of-Edge framework that combines DP and blockchain to enhance privacy and energy efficiency in IoT systems, which was validated through experimental evaluations of Ethereum. Their energy-efficient design improves privacy protection without compromising performance. The authors of [106] proposed a DP fog-computing-oriented approach for governmental data publishing that incorporates a MaxDiff histogram algorithm to protect citizens' privacy while improving data utility. Their approach effectively mitigated privacy vulnerabilities by adding Laplace noise to the dataset and optimizing the data bins based on frequency differences. In [107], the researchers presented a novel local DP algorithm designed to address privacy issues in deep-learning IoT applications by incorporating a randomization layer before data transmission. The authors of [108] explored the protection of medical information in a digital era. They introduced a model that integrates k-anonymity with DP to enhance security. In [109], the authors examined how incorporating second-tier data into the loss function can enhance DP in convex optimization. They showed that their proposed method achieves quadratic convergence and effectively minimizes excess loss for highly

convex loss functions. In [110], it was observed that the bias-variance trade-off becomes more significant in differentially private learning scenarios when users are allowed to submit multiple examples, owing to the limitations imposed by the contribution-bounding datasets. In [111], the authors investigated how noisy gradient descent algorithms affect DP during training. They demonstrated that the Rényi divergence between models trained on similar datasets has a tight bound, and for smooth, strongly convex loss functions, privacy loss decreases rapidly. In [112], the authors introduced a data aggregation method for edge computing that combines DP with secret sharing. They employed Gaussian noise and a two-layer aggregation approach to reduce the probability of privacy breaches. In [113], the authors tackled the challenge of privacy in edge computing by developing a location data collection method based on local DP. Their approach uses Voronoi diagrams to divide the road network into regions, then applies randomized perturbation to obscure users' exact locations. This method strikes a balance between privacy protection and data utility. In [114], the researchers focused on privacy-aware video streaming in mobile edge computing. Their system dynamically adjusts privacy levels based on real-time conditions, ensuring that users' location and usage patterns remain private. Their online learning-based algorithm optimized video frame rate, resolution, and offloading decisions, reducing latency and energy consumption without compromising performance. In [115], researchers proposed a privacy-preserving framework that effectively connects individuals and their virtual twins and integrates DP, multitask learning, and blockchain to address the influence of diverse environments. They also introduced a validation mechanism based on model quality to ensure precise and legitimate updates to models in a virtual setting. In [116], the authors created Time Intervals via DP and Frequency Vectors to ensure the privacy of the time interval data. They also introduced an algorithm for optimizing maximum likelihood estimation using these vectors and space partitioning to improve privacy. Drawing from our review of the selected DP studies, Table 5 presents a comparison of their strengths and limitations based on the criteria of our structured evaluation framework, including privacy, security, scalability, and maturity.

C. TRUSTED EXECUTION ENVIRONMENT (TEE)

A TEE refers to a secure zone inside a processor that protects both the data and the program being used, especially during tasks such as model training and computing. The primary objective of TEEs is to strengthen data confidentiality and integrity by keeping everything hidden from external access. Typically, data are encrypted when sent and decrypted within the TEE. Software and hardware solutions work together to ensure data safety [117]. Applications that operate inside a TEE are called Trusted Applications (TAs) and operate in a much more secure environment than those operating in regular operating systems. This additional security layer guarantees that the Trusted Operating System (TOS) and TAs are significantly more reliable than the general-purpose software environments [118].

The concept of a TEE was first formally established by the Open Mobile Terminal Platform (OMTP) in its "Advanced Trusted Environment (OMTP TR1)" standard [119]. Azure Confidential Computing, introduced in 2020, is a well-known example of real-world TEE. This enables customers to move existing workloads to Azure without the need to change any code or experience performance issues. This service offers two types of workloads: enclave-based and lift-and-shift workloads. The enclave-based option uses Intel Software Guard Extensions to create a protected memory area, known as an Encrypted Protected Cache, within a virtual machine. The lift-and-shift option allows organizations to quickly transition their existing workloads to the cloud with minimal disruption while retaining flexibility for future cloud optimization. New technologies such as modern Graphics Processing Units (GPUs) also support the TEE features. For example, NVIDIA introduced products that support secure AI applications using GPU TEE solutions.

TEEs rely on hardware isolation, in which only TAs can access the full processor, memory, and peripherals of the device. Regular applications such as those installed by users cannot interact with these resources. To prevent tampering, a "hardware root of trust" is embedded in the chip during the manufacturing stage. This involves private keys stored in one-time programmable memory to ensure that these keys cannot be changed, even if the device is reset. Public versions of these keys are stored in a manufacturer's database, and only software signed with a trusted party key can access critical system features. When a TA is loaded into memory, it undergoes a process called attestation. This guarantees that the TA has not been tampered with. A server provides a cryptographic "nonce" (a random number used only once) to verify the integrity of the application. Importantly, faking this process with simulated hardware is impossible without access to the private keys embedded in the hardware of the device, which is unique to each piece of hardware [120], [121]. However, this strict attestation process incurs additional operational costs. Many TEEs rely on remote attestation services that require trusted third-party infrastructure to verify enclave integrity. This dependency not only increases the cost of deployment but also raises concerns regarding vendor lock-in, as organizations must often rely on a single hardware manufacturer's ecosystem to maintain TEE security.

Trust in a TEE includes all elements involved, from the code to the underlying TOS and supporting infrastructure. Every step in setting up and running a TEE follows a strict process beginning with the read-only memory (ROM) boot stage. Only the verified code is allowed to run, and TAs can only access their data. No one TA can access the assets of another TA to ensure strong isolation between them. Inside a TEE, TAs need only trust the TOS and not worry about the

TABLE 5. Summary and Comparison of recent significant DP approaches.

Ref	Privacy	Security	Scalability	Maturity	Description	Pros	Cons
[102]	✓	×	×	Prototype	A data collection framework based on edge computing, where raw data is processed with DP on edge servers.	Enhances privacy protection by ensuring original data is not retrievable even if leaked; reduces communication and storage costs.	Dependence on edge servers may create scalability challenges and introduce new points of failure.
[103]	✓	✓	×	Prototype	A DP-based sustainable fog computing query model for data centers that quantifies privacy-preserving quality through mathematical proof.	Effectively resists various privacy attacks while achieving high data utility; flexible to device heterogeneity.	May require computational resources for real-time queries, impacting performance under heavy load.
[104]	✓	✓	✓	Experimental	A local classification algorithm for data centers that adds DP mechanisms to handle sensitive information during data mining.	Higher iteration efficiency and better security with reliable privacy protection characteristics.	Performance may vary significantly based on data center heterogeneity and data distribution.
[105]	✓	✓	×	Prototype	A blockchain-enabled Internet of Edge framework integrating DP for a scalable, privacy-preserving system.	Improves privacy protections without compromising performance; energy-efficient design.	Complexity of integrating multiple technologies can lead to implementation challenges and potential delays.
[106]	✓	✓	×	Prototype	A fog-computing-enabled DP model for governmental data publishing to protect citizens' privacy against potential vulnerabilities.	Effectively prevents privacy disclosure even with strong background knowledge from adversaries.	Implementation may be limited by computational resources and may not scale well under high query volumes.
[107]	✓	×	×	Prototype	A local DP algorithm that adds a randomization layer in a convolutional neural network architecture for preserving privacy in deep learning.	Maintains high accuracy with low privacy budgets while enhancing practical utility for IoT-driven environments.	Requires significant computational resources and may be challenging to implement in highly constrained IoT devices.
[108]	✓	✓	✓	Concept	An algorithmic model incorporating both DP and k-anonymity ensured minimal risk of privacy breaches.	Can effectively reduce privacy leakage and prevent information security breaches.	Lacks proper safeguards for node security and remains in the simulation phase, with no operational testing carried out yet.
[109]	✓	×	×	Concept	Introduces an enhanced version of Nesterov and Polyak's regularized cubic Newton method and presents a second-order DP algorithm for unconstrained logistic regression.	Illustrates that second-order techniques are applicable in the DP setting both for strengthening worst-case convergence assurances and developing faster practical algorithms.	The expense of constructing and inverting the Hessian becomes unmanageable when the diameter is very large.
[110]	✓	×	✓	Concept	Investigates the clipping bias-variance trade-off, concluding that it is a fundamental aspect of DP learning.	Users can adjust contribution limits according to the data's statistical properties utilizing boundary conditions.	Increased noise leads to model accuracy and efficiency drops.
[111]	✓	×	×	Concept	Enables noise gradient descent algorithms to be examined by considering the dynamics of privacy loss.	The information leakage rate during training is analyzed, revealing significantly tighter bounds compared to composition-based methods.	The problem lies in extending this analysis to cover non-convex, non-smooth loss functions, as well as stochastic gradient updates.
[112]	✓	✓	✓	Prototype	Employs a sensor fog-cloud architecture along with DP and additive homomorphic encryption.	Requiring only a single round of data exchange between the smart meter, its connected Fog node, and the Cloud.	Adding more noise results in a decline in both model accuracy and efficiency.
[113]	✓	×	✓	Prototype	Proposes a local differential privacy-based mechanism for data collection in edge computing, using Voronoi diagrams and randomized perturbation.	Reduces latency and improves data processing efficiency in IoT environments.	Relies on the assumption that edge nodes behave honestly, which may pose risks in adversarial settings.
[114]	✓	✓	×	Prototype	Introduces a personalized and privacy-aware video stream offloading scheme in mobile edge computing.	Dynamically adjusts privacy levels based on real-time constraints, balancing accuracy, energy efficiency, and security.	Increased computational complexity due to the optimization process may lead to higher latency in resource-constrained devices.
[115]	✓	✓	×	Prototype	A connectivity approach for human-to-virtual twins that incorporates DP, multi-task learning, and blockchain for secure, privacy-preserving, and efficient communication.	Accelerates learning while maintaining accuracy and privacy, and minimizing communication costs. Ensures authorized model evolution.	May face challenges with scalability when dealing with larger, more complex systems due to heterogeneous environments.
[116]	✓	×	✓	Prototype	Enhances the release of time intervals under DP using a partitioning technique for frequency vectors.	Balance noise and structural errors.	Sensitive queries to a differentially private database may produce incorrect conclusions.

presence of other TAs. For example, if one TA creates a file named "Secrets", another TA can create a file with the same name. However, these are treated as completely separate. The files did not interfere with each other. Additionally, no application in a regular operating system can access the files created by TAs. An adversary cannot move a TA's assets between devices because all the TEE storage is tied to the original device [122].

Many cloud-native workloads rely on seamless data sharing and interoperability between virtualized environments, which TEEs inherently restrict. As a result, TEEs are often unsuitable for highly parallelized tasks that require frequent communication between instances. Furthermore, performance overhead associated with switching between secure and non-secure execution modes can degrade system responsiveness, particularly in high-throughput applications. Although TEEs can be tested in laboratories, this level of testing is rare because of its high cost. Instead, most users rely on attestation services to prove the trustworthiness of TEE. If any signature check failures occur, the TEE does not function. Therefore, users must trust that the device manufacturer has properly designed the TEE. The fundamental hardware isolation of TEEs creates scalability bottlenecks, as individual instances must independently manage security states, increasing overhead when deployed across multiple nodes. Additionally, as TEEs evolve, backward compatibility becomes a concern. Organizations investing in current-generation TEEs must ensure that future hardware releases maintain compatibility with existing applications, which is not always guaranteed. This poses a risk to long-term adoption, particularly in enterprise environments with extensive legacy infrastructure.

In recent years, governments, companies, and CSPs have increasingly used TEEs to manage sensitive information securely. For instance, the Memory Safe Trusted Execution Environment is an open-source secure computing platform developed by Baidu and Intel in 2018. This platform is designed with privacy at its core, using Rust for memory safety and Intel SGX for secure execution. It comes with built-in tools for machine learning and cloud computing, simplifying complex tasks and helping developers efficiently create new SGX applications using dedicated development toolkits. Much other outstanding research has also been conducted to secure DCC with the help of TEEs. To safeguard against leakage of sensitive data, in [123], the authors introduced a detailed access-control method utilizing attribute-based encryption in conjunction with a TEE under ciphertext policies. In [124], a trusted IoT architecture was presented for IoT systems that incorporate TEEs and various security measures. They used hierarchical Colored Petri Nets for model-based testing to examine essential security aspects, such as communication and encryption. The study also involved creating a formal security model and employing model checking to ensure alignment with expected functionality. In [125], the authors suggested a novel trust zone structure that provides the essential components of the TEE to enhance the security of edge devices. In [126], the authors presented a privacy protection strategy that emphasizes aggregation based on individual customers. The TEE handles critical operations including critical distribution, data decryption, load monitoring, billing, and additional related services. Instead of using a central aggregator, this approach utilizes aggregation functions customized for specific customers over a designated timeframe. In [127], the authors introduced a secure access method for mobile devices using TEE. They designed a multi-environment framework known as open TEE to create a trustworthy environment that is separate from a standard rich environment. Their approach includes mechanisms such as file slicing and authorization checks to protect the confidentiality of private files. In [128], the researchers described a TEE architecture for neural processing units aimed at improving security and addressing vulnerabilities. It uses tile-based translation and verification to ensure strong isolation and reduce the memory check overhead. To address the security challenges in IoT, [129] developed an IoT data-sharing framework that combines TEEs with blockchain technology. This architecture incorporates both the on- and off-chain strategies. The authors of [130] addressed problems such as imprecise granularity, insufficient audit capabilities, and inadequate process management in IoT access control. They developed a solution using blockchain and currency-based access control models that incorporated TEE technology to enhance privacy protection in IoT systems. Likewise, [131] introduced a lightweight TEE for in-storage computing that addresses security concerns in modern solid-state drives by isolating programs from flash management functions. In [132], the authors proposed a novel framework that leverages the TEE of an edge device to limit the vulnerability surface of Deep Neural Networks. The more sensitive layers are executed inside the TEE by partitioning the model layers, whereas the less critical layers are run in the untrusted operating system. The authors of [133] focused on privacy concerns in IoT devices that collect sensitive personal and behavioral data. They proposed a generic data-aggregation scheme utilizing TEEs to guarantee data privacy in cloud environments. Their method accommodates diverse data types and executes sophisticated computations such as machine learning and deep learning algorithms with strong privacy guarantees. In [134], the authors proposed a TEE-based architecture to enhance the security and privacy of cloud-hosted cyber-physical systems. The paper presents an implementation setup validated through a testbed system. The authors of [135] introduced a blockchain-based IoT data-sharing framework that integrates ciphertext-policy attribute-based encryption with TEE. Their approach enables efficient policy updates and attribute revocation, reducing the computational overhead associated with traditional access control mechanisms. In [136], the authors proposed a distributed TEE architecture designed for secure interactions between multiple trusted devices. Their framework enables seamless

TABLE 6. Summary and Comparison of recent significant TEE approaches.

Ref	Privacy	Security	Scalability	Maturity	Description	Pros	Cons
[123]	✓	✓	×	Experimental	Ciphertext-policy attribute-based encryption has been applied to enhance security by enabling fine-grained access control and supporting critical operations in a TEE environment.	Minimizes vulnerabilities in sensitive data, reducing risks associated with a single authority.	Time overhead; lack of efficiency in access control.
[124]	✓	✓	×	Prototype	Security features related to both acceptable and restricted behaviors are represented using hierarchical Colored Petri Nets, facilitating model-based testing.	Demonstrates the ways in which the TEE strengthens security and privacy.	Lacks performance and scalability.
[125]	✓	✓	×	Prototype	A trust zone-based architecture that provides the TEE's essential components to enhance edge devices' security.	As a result of this trusted zone, devices can communicate in a completely secure environment.	Requires external hardware resources.
[126]	✓	✓	×	Experimental	A privacy-preserving method for smart grid based on TEE.	Ensures data accuracy and protects against false data injection.	Limited scalability; updating meters affects aggregation operations, especially with selected customer aggregators.
[127]	✓	✓	×	Prototype	An access method based on a TEE designed to protect the integrity of private files.	Satisfies requirements for both privacy and security.	Limited stability and resources; relies heavily on trust; requires enhanced client efficiency for accessing multiple files simultaneously.
[128]	✓	✓	×	Prototype	A TEE architecture for neural processing units is designed to enhance security and address vulnerabilities through the use of tile-based translation and verification methods.	Runtime costs for security checks are reduced effectively.	Lack of transparency; challenges in handling resource-intensive tasks.
[129]	✓	✓	×	Prototype	A novel TEE-blockchain-powered data sharing framework.	Enhanced performance compared to centralized methods.	Lacks scalability and interoperability.
[130]	✓	✓	×	Experimental	A TEE-supported access control model based on encrypted currency and blockchain.	Precise control, strong oversight, and access process management.	Potential overhead due to blockchain integration; May be complex to implement in large-scale IoT networks.
[131]	✓	✓	×	Prototype	A lightweight TEE for in-storage computing that addresses the security concerns in modern solid-state.	Minimal hardware cost; maintains performance benefits of in-storage computing.	Requires additional hardware resources for trust zone extensions.
[132]	✓	✓	×	Experimental	A novel framework leveraging an edge device's TEE to limit the vulnerability surface of Deep Neural Networks.	Provides model privacy with only 3-10% performance overhead.	Limited by the memory of the edge device's TEE, requiring careful partitioning of model layers.
[133]	✓	✓	×	Experimental	A data aggregation framework to protect the privacy of heterogeneous IoT data.	Supports heterogeneous data processing and complex computations, ensuring privacy in cloud-based aggregation.	Data must be extracted from each party's site and consolidated at the data center where the TEE is located, a scenario frequently encountered in cloud computing.
[134]	✓	✓	×	Prototype	A TEE-based architecture for securing cloud-hosted cyber-physical systems.	Reduces deployment costs and enhances system resilience.	Performance overhead due to TEE integration; limited scalability in large-scale deployments.
[135]	✓	✓	×	Prototype	A TEE blockchain-based IoT data-sharing scheme.	Efficient policy updates and revocation processes.	Complexity in managing cryptographic operations; limited scalability and widespread adoption.
[136]	✓	✓	×	Prototype	A distributed TEE architecture enabling secure interactions across multiple trusted devices.	Provides seamless digital transactions with lightweight secure channels.	Complexity in managing multiple TEEs; requires extensive integrity verification.

communication between heterogeneous TEEs, allowing interconnected wearable devices, such as smartwatches and smartphones, to conduct secure transactions collaboratively. The study also includes a proof-of-concept implementation based on the European Digital Identity wallet, demonstrating its feasibility for electronic identification applications. Table 6 presents a detailed comparison of relevant studies incorporating TEE, based on our structured evaluation framework.

D. FEDERATED LEARNING (FL)

FL was developed by Google in 2016 to address rising concerns regarding data abuse and communication costs [137], [138]. In today's world, with growing concerns over personal

information usage, FL has emerged as a critical tool for safeguarding data privacy. By keeping raw data stored locally on participant devices, FL enables collaborative learning and computation without the need to centralize sensitive information.

Integrating FL into a DCC involves utilizing edge or cloud servers that periodically collect the trained parameters to refine the global model. This global model is subsequently returned to the edge devices for additional local training to ensure that the sensitive data are safe and decentralized. The FL training process consists of five essential stages. First, the FL server selects a model tailored to the local data available from clients. Next, a group of clients is selected randomly or through selection algorithms that optimize client selection based on the data quality or device performance criteria. The server then broadcasts the updated or original global model to the chosen clients who download the model parameters for local training. Upon finishing local training, every client transmits its updates to the server. Finally, the server aggregates these updates to create a new global model without accessing client data. This iterative cycle continued until the model fulfilled the accuracy criteria.

Suppose N clients each hold their respective training datasets $D_1, D_2, ..., D_N$. The aforementioned iterative process runs to refine the global model M_{glob}. The performance metric, P_{sum}, derived from training the aggregated model M_{sum}, serves as the baseline. The evaluation metric for the FL-trained model M_{glob} is represented as P_{glob}. Accuracy is typically used as the core evaluation metric, where the difference between P_{sum} and P_{glob} is small, denoted by $|P_{\text{sum}} - P_{\text{glob}}| < \delta$, with δ being a non-negative real number. A smaller δ signifies that the FL model operates closer to the centralized benchmark. The objective function for FL can be mathematically represented as:

$$\min_{w} F(w) = \sum_{j=1}^{N} q_j F_j(w), \tag{3}$$

where:

$$F_j(w) = \frac{1}{m_j} \sum_{i=1}^{m_j} \mathcal{L}(x_i, y_i; w), \tag{4}$$

with $q_j \geq 0$ and $\sum_j q_j = 1$. The term w represents a single global model and q_j specifies the relative contribution of each client, with common choices for q_j being $q_j = \frac{1}{m}$ or $q_j = \frac{m_j}{m}$, where m_j is the sample size for the j-th client and m is the total combined sample size across all clients. $F_j(w)$ expresses the local objective function for the j-th client, where x is the data feature, y is the corresponding label, and $\mathcal{L}$ is the loss function. The main purpose of FL is to optimize this objective function while reaching a consensus on model weights among all participating clients [139]. By achieving this goal, FL minimizes the bandwidth and time needed for inference and training. Since local data is stored on user devices and is rarely shared with remote servers, the updated model can make predictions directly on users' devices. Thus, this design enhances privacy and security and promotes efficient collaborative learning that consumes less power.

Scheduling optimization techniques in FL are generally categorized as either **synchronous** or **asynchronous** [140], [141]. Figure 6 illustrates the synchronous and asynchronous workflow of FL. In synchronous communication, a selected group of clients is assigned to train local models during each training round. However, inconsistencies in device performance or network stability can lead to some clients failing to respond within the expected timeframe. When this happens, the server must wait until a sufficient number of responses are received. If this threshold is not met, the server discards the round and proceeds to the next iteration. On the other hand, asynchronous communication operates differently by allowing clients and the server to interact without waiting for all responses to arrive at once. This is particularly beneficial when distributed across multiple edge devices, as it enhances convergence speed. In asynchronous optimization, FL participants can transmit gradient updates to the central server immediately after completing each local update, a feature not feasible in synchronous FL. It is less affected by variations in client resources, making it a more reliable option in heterogeneous environments [142].

FL can be structured into three primary configurations: **Cloud-Enabled**, **Edge-Enabled**, and **Hierarchical (Client-Edge-Cloud-Enabled)**. The choice of configuration depends on the specific application and infrastructure available, as well as the performance requirements. Figure 7 demonstrates these three configurations.

In cloud-enabled FL, the global learning process relies on geographically dispersed edge devices that communicate with a central cloud server [143]. Clients, such as IoT devices, collaborate by training local models on their data and sending updates to the cloud for aggregation. The cloud server consolidates these updates into a global model, which is then redistributed to the clients for further local training. Although cloud-enabled FL can scale across millions of clients and cover large geographic areas, it faces challenges related to communication costs and network congestion. Large model updates transmitted between clients and the cloud can slow the learning process. Moreover, frequent communication between devices and the cloud requires substantial bandwidth and may result in performance degradation under heavy traffic conditions.

In edge-enabled FL, learning is localized because nearby devices interact with a local edge server to compute the global model. Every device performs local training and sends the results to the edge server, aggregating the updates and refining the global model before redistributing them to the device. This configuration reduces communication latency owing to the proximity between the devices and the edge server, making it ideal for real-time applications, such as autonomous driving or smart cities [144]. However, edge-enabled FL is restricted by the limited computational resources of edge servers, which are less powerful than cloud

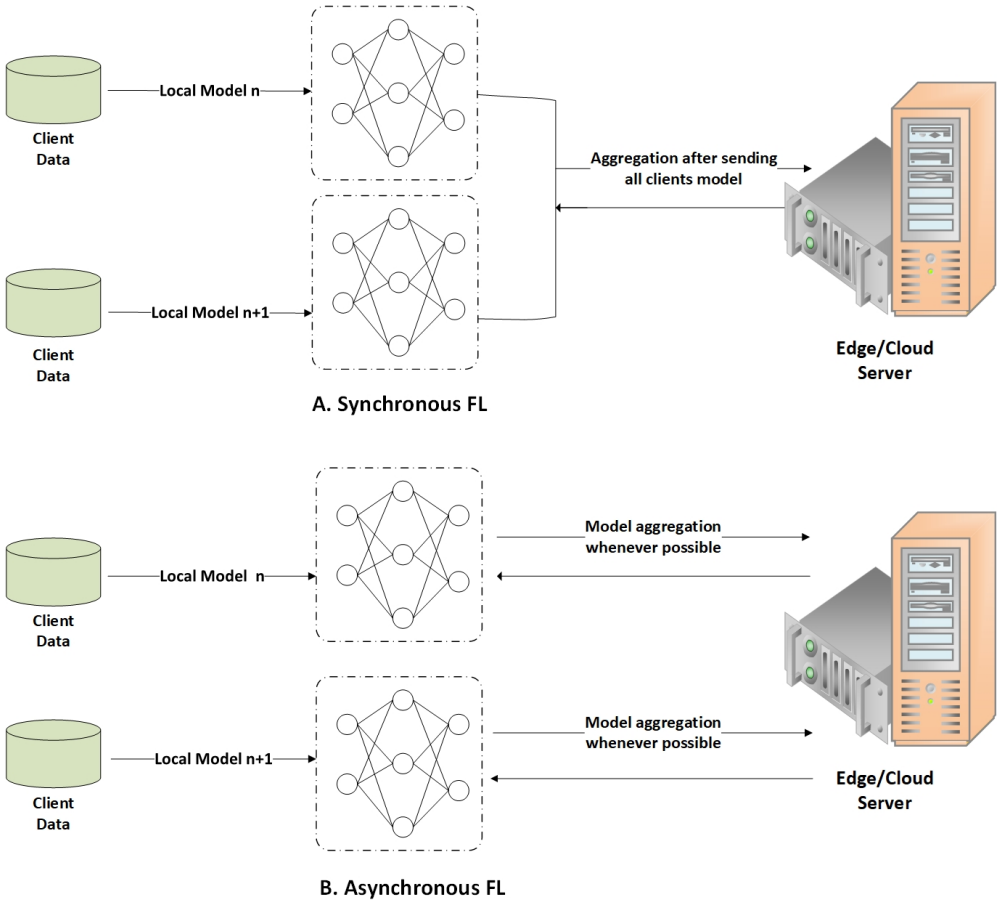


FIGURE 6. **Synchronous and asynchronous workflow of FL.**

servers. This limits the scale of computing tasks that can be handled and the volume of devices that can connect to an edge server.

Hierarchical FL combines the strengths of both cloud- and edge-enabled configurations. Clients first communicate with their local edge servers, which perform a preliminary aggregation of the local model updates before sending them to the cloud. The cloud server aggregates these high-level updates to produce the global model. This setup minimizes direct communication between clients and the cloud, reduces network congestion, and improves efficiency. Hierarchical FL also helps manage geographical distribution effectively, as nearby clients can optimize local models at the edge level before contributing to the global model [145]. This structure is particularly beneficial for reducing cross-region communication and ensuring region-specific accuracy during the learning process. By involving both edge and cloud resources, hierarchical FL balances scalability, efficiency, and privacy [146], [147].

A significant feature of FL is its robust privacy guarantee, as private data never leaves local devices. These guarantees can be reinforced using three dominant techniques: **Data Encryption**, **Data Perturbation**, and **Anonymization** [10], [17]. Data encryption is crucial for securing communicated parameter updates. For instance, *Homomorphic Encryption* is a widely used data encryption technique that allows computations to be processed with encrypted data, meaning that user data remain confidential even while being processed. This technique ensures that sensitive information is never exposed during training because only encrypted values are exchanged. A case study of this technique is Federated AI Technology Enabler an open-source project launched by WeBank in 2019 to support the FL ecosystem. It provides a distributed secure computing platform that integrates homomorphic encryption and hash functions to facilitate collaboration between multiple parties while maintaining compliance with privacy regulations [148]. In addition to encryption, FL employs data perturbation techniques to enhance privacy. For example, *Additive Masking* involves introducing noise to the gradient updates sent by clients, obscuring sensitive information, while still allowing the server to aggregate useful insights. This method protects client details by ensuring that the server sees only the aggregated information to minimize the risk of identifying individual users. Although data perturbation techniques offer robust privacy protection, they often result in a decline in data utility. Anonymization has been introduced as a solution to protect user identities by ensuring that their

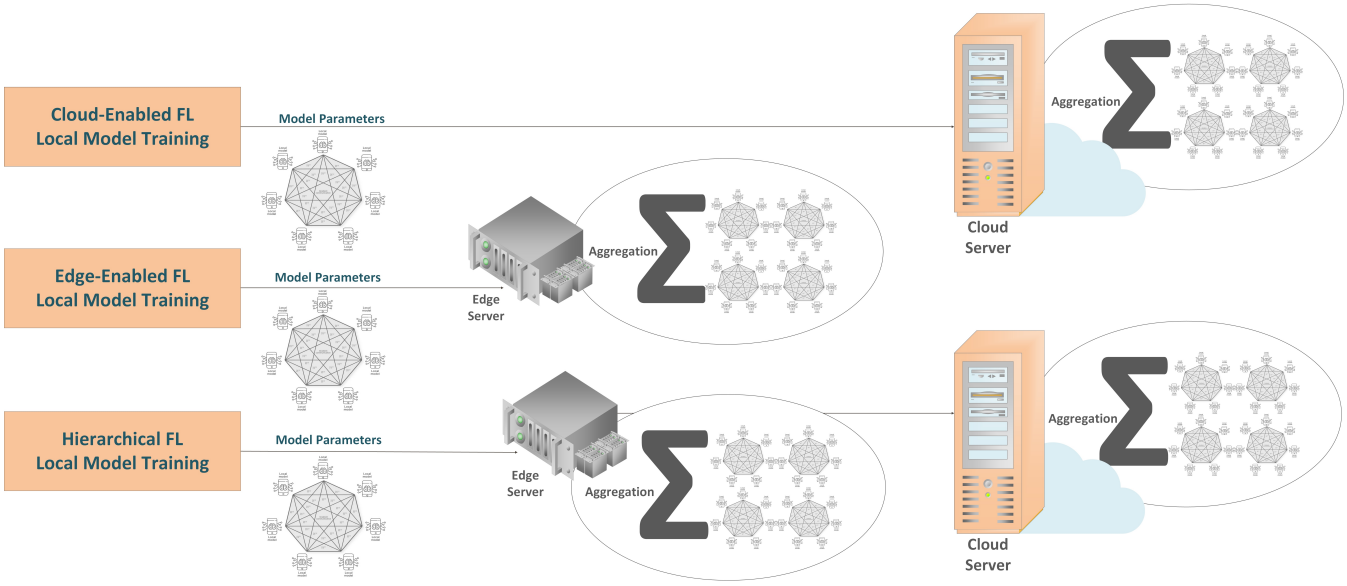


FIGURE 7. **Cloud-Enabled, Edge-Enabled, and Hierarchical FL configurations.**

contributions cannot be linked to them during model updates. It removes identifiable information, aggregates user data to prevent isolation, and may use methods such as randomized communication or mixed networks to obscure the origin of the updates. This approach enhances privacy by ensuring compliance with privacy regulations. However, this can increase computational complexity.

Another key aspect of FL is its robust security, which can be ensured through three major techniques for identifying malicious updates: **Anomaly Detection**, **Statistical Information**, and **Blockchain** [149], [150]. Anomaly detection identifies disruptions caused by external threats such as internal issues and server malfunctions. By analyzing patterns in incoming updates, anomaly detection can flag unusual behaviors that deviate from expected norms, thus safeguarding the training process from potential disruptions. Moreover, leveraging statistical information can be an effective method for mitigating the impact of malicious activities. For instance, *Geometric* and *Coordinate Medians* can aggregate updates to minimize the influence of outliers and malicious contributions. Finally, blockchain technology can further promote authentic contributions and enable reliable tracking of updates. A transparent ledger can verify each update to ensure that only legitimate contributions are considered in the model-training process. By utilizing decentralized consensus protocols, blockchain addresses issues related to single points of failure and possible malicious activities in the FL.

Recently, researchers have developed numerous exciting privacy-preserving and secure FL frameworks for DCC. In [151], an FL framework was introduced to enhance Block Hunter detection with minimal bandwidth usage in blockchain-enabled Industrial IoT (IIoT). This system uses a cluster-based structure for anomaly detection by integrating several learning models in a decentralized setting. In [152], the authors addressed privacy vulnerabilities in next-generation IoT environments using a privacy-preserving FL model. Their approach combines synchronous and asynchronous FL modes, utilizing homomorphic encryption to safeguard sensitive information. The authors of [153] explored anonymization techniques to address user-level privacy vulnerabilities and introduced a Generative Adversarial Network (GAN)-based FL framework incorporating a multitask discriminator. In [154], a novel FL architecture was proposed to detect malware in IIoT environments by focusing on Android applications. The architecture employs a GAN to enhance the defense mechanisms and ensure robust collaboration. In [155], the authors presented a privately enhanced FL model to secure Industrial Artificial Intelligence (IAI) applications. Their non-interactive model prevents data leakage, even in the event of collusion between entities, with experimental results showing its superiority in both accuracy and efficiency over traditional approaches. In [156], an FL anomaly detection system for smart electric grids was presented, combining Long Short-Term Memory (LSTM) networks and autoencoders with Median Absolute Deviation and Mean Standard Deviation techniques to improve detection accuracy. The use of FL guarantees the privacy of the critical energy data. In [157], an FL and blockchain-based approach was proposed to preserve the privacy of electronic health records. In [158], a federated deep reinforcement-learning framework was developed to address privacy issues related to task offloading in distributed cloud environments. The framework manages context-aware data at different system levels—CloudAI, EdgeAI, and DeviceAI—to ensure privacy during the task execution process. In [159],

the Light-SecAgg method was introduced to enhance the performance of secure FL aggregation by using mask coding and decoding. This method minimizes the dropout effects and improves scalability by combining model training with on-device encoding. Furthermore, the modular architecture and parallel processing of the system enhance the efficiency of chunked mask management. In [160], the authors designed a secure FL framework with optimized algorithms that could effectively withstand Byzantine failures in DCC systems. Their work emphasizes achieving the best possible statistical performance, making it a reliable solution for real-world applications. The authors in [161] developed an FL approach that guarantees privacy protection and verification. To maintain privacy while minimizing both computational and communication costs, their method incorporates Chinese remainder theorems along with homomorphic encryption. The authors of [162] discussed the integration of FL into IoT-based healthcare systems, utilizing data encryption methods, such as homomorphic encryption, to safeguard local data. In [163], researchers focused on blockchain-based FL and introduced a reputation-based approach to encourage data owners to contribute high-quality data. They implemented a reward distribution system as an incentive mechanism for the FL process. The authors of [164] introduced a privacy-preserving FL approach for DCC to enhance security and efficiency. In their approach, fog nodes were used to collect data, addressing challenges such as uneven data distribution and differences in computational power among users. They utilized homomorphic encryption combined with blinding to protect the model's security. In [168], the authors introduced a non-interactive, privacy-preserving FL framework for DCC that leverages a dual-server architecture. This approach enhances both system security and efficiency and shows resilience to client dropouts. The model maintains a communication burden on client-server interactions that is limited to no more than twice that required for plaintext processing. Finally, in [169], an FL-based anomaly detection system was introduced to secure the IoT networks. This approach preserves user data privacy by training Gated Recurrent Units models on-device and sharing only learned weights with a central server. Table 7 compares recent major FL studies according to our structured evaluation framework.

VII. DISCUSSION

Building on the comprehensive review presented in Section VI, this section addresses the AQs raised in the Introduction. It first provides a detailed summary of the performance and efficiency of the four generations of PPC (i.e., MPC, DP, TEE, and FL) within real-world DCC applications that emphasize how each generation contributes to ongoing development. To address AQ2, we discuss the challenges in implementing FL frameworks in DCC environments and propose strategies to overcome them. We also assess the strengths and limitations of new paradigms for enhancing FL security and privacy, including secure FL based on malicious updates and privacy-preserving FL techniques involving data encryption, perturbation, and anonymization. Finally, we explore how FL can balance security, performance, accuracy, and efficiency using these methods in the DCC context.

AQ1: What are the key advantages of FL over MPC, DP, and TEE when addressing privacy and security challenges in DCC?

FL stands out as a particularly effective solution for managing large datasets in distributed environments. While MPC, DP, and TEEs offer unique privacy advantages, they often face scalability and efficiency challenges in large-scale DCC contexts. FL's decentralized architecture enables it to maintain user privacy while meeting the scalability and performance demands of modern distributed cloud applications. This makes FL a more adaptable approach to PPC in today's data-driven landscape, supported by the following five key strengths [165], [166], [167]:

1) **Decentralized Privacy Model:** FL stands out for its unique distributed design, which minimizes the potential for unauthorized access by keeping user data on-device, transmitting solely model updates to the server. This is a significant departure from the process used in MPC, where raw data must be split among various parties and securely computed together, which can be complicated for large-scale applications. By contrast, DP works by adding random noise to data or query results to prevent sensitive information from leaking during analysis. However, if queries are repeated or additional information is available, the privacy assurance provided by the DP can weaken. This makes DP less effective in DCC environments, where sensitive data are frequently accessed. Similarly, TEEs secure data within hardware-protected environments; however, they rely on extracting data from various devices to a central location, which can be cumbersome in distributed cloud settings.
2) **Resilience to Non-IID Data:** In real-world DCC scenarios, user data are usually not uniform. Although non-IID data may occasionally lead to biased FL models, FL is generally better equipped to handle such data through adaptive techniques, such as federated averaging adjustments, compared to other PPC methods. With FL, model training can occur directly on user devices, allowing the models to learn from the specific characteristics of the data of each user. This localized training means that the FL can produce models that reflect user- or location-specific insights. In contrast, MPC and TEEs are not designed to address data distribution issues; instead, they primarily rely on data splitting or encryption for centralized processing. This focus implies that the non-uniformity of updates from non-IID data can lead to suboptimal model convergence and performance degradation. Additionally, DP suffers from non-IID data, because adding noise

TABLE 7. Summary and Comparison of recent significant FL frameworks.

Ref	Privacy	Security	Scalability	Maturity	Technique	Description	Pros	Cons
[151]	✓	✓	✓	Prototype	Anomaly Detection	An FL framework for minimum bandwidth Block Hunter identification using cluster-based architecture in IIoT networks.	High accuracy in detecting anomalous activities while requiring minimal bandwidth.	Does not specify how to handle non-IID data or user dropout issues.
[152]	✓	✓	✓	Experimental	Data Encryption	A privacy-preserving FL model in IoT environments by combining synchronous and asynchronous modes with homomorphic encryption.	Addresses user dropout and low-quality data; high functionality and accuracy; low system overhead.	The involved clients may be unreliable since they commonly rely on battery-powered systems and less powerful communication media.
[153]	✓	✓	×	Experimental	Anonymization	A GAN-based FL framework that employs anonymization techniques to address user-level privacy vulnerabilities.	Effective anonymization techniques to address user-level privacy leakage; novel use of multi-task GAN-AI for fine-grained privacy preservation.	Large-scale updates can cause bandwidth and latency issues, with efficiency based on communication channels.
[154]	✓	✓	✓	Prototype	Anomaly Detection	An FL architecture for malware detection in IIoT utilizing GANs to bolster defense mechanisms.	Robust malware detection with high accuracy.	Focuses primarily on Android malware and performance in other types of malware defense not addressed.
[155]	✓	✓	×	Prototype	Data Perturbation	A non-interactive privacy-enhanced FL model securing IAI applications, preventing data leakage even with collusion among entities.	Strong privacy protection against data leakage and collusion, efficiency in IAI applications.	May not fully scale for extremely large networks due to non-interactive model parameters.
[156]	✓	✓	×	Prototype	Anomaly Detection	An FL anomaly detection system for smart electric grids, integrating LSTM networks and autoencoders with statistical and encryption techniques.	High accuracy and low computational cost in anomaly detection for smart grids.	Communication overhead may limit scalability in large smart grid deployments.
[157]	✓	✓	×	Prototype	Blockchain	A secure FL framework based on identifying malicious updates through blockchain to enhance the security of electronic health records.	High accuracy and performance in malicious updates classification.	Blockchain can introduce overhead in terms of bandwidth and latency.
[158]	✓	✓	×	Prototype	Anonymization	A privacy-preserving deep reinforcement FL framework for DCC environments, managing context-aware data across multiple system levels.	Ensures high-context privacy on local devices while offloading tasks; improves scheduling efficiency through context-aware management.	Increased complexity in managing different levels of context-awareness; performance might vary depending on task types and offloading requirements.
[159]	✓	✓	✓	Experimental	Data Perturbation	The Light-SecAgg approach improves secure FL aggregation performance through mask coding and decoding.	Enhances scalability and minimizes dropout effects; reduces training time while maintaining privacy as leading approaches.	Requires retransmission of incorrect gradients due to noise masking unmasked gradient values.
[160]	✓	✓	×	Experimental	Statistical Information	A secure FL framework designed to withstand Byzantine failures, optimizing algorithms for statistical performance.	Robust against Byzantine failures; achieves optimal statistical performance with strong error rate guarantees for convex loss functions.	Single-round median-based distributed algorithms may face limitations in complex environments with non-convex functions.
[161]	✓	✓	✓	Prototype	Data Encryption	An FL approach ensuring privacy protection and verification, utilizing Chinese remainder theorems and homomorphic encryption.	Detects malicious behavior on aggregate servers; reduces communication and computation costs while enhancing efficiency.	Public key misuse by the server could reduce model accuracy by extracting information from shared gradients.
[162]	✓	✓	×	Prototype	Data Encryption	An FL framework for IoT-based healthcare systems, employing homomorphic encryption for safeguarding local data privacy during model training.	Effectively meets users' privacy and security needs, with detailed security analysis confirming its reliability.	Does not address challenges related to heterogeneous clients with limited hardware or asynchronous FL, which affects overall efficiency.
[163]	✓	✓	×	Prototype	Blockchain	A reputation-based approach in blockchain-based FL, motivating data owners to contribute high-quality data through a reward distribution system.	Significantly improves high-quality model aggregation in FL while safeguarding the training process from interference by malicious nodes.	Dependence on the blockchain infrastructure may introduce scalability challenges in large-scale deployments.
[164]	✓	✓	✓	Experimental	Data Encryption	A privacy-preserving FL scheme for fog computing, utilizing homomorphic encryption and blinding.	Efficient against collusion by multiple malicious entities; addresses uneven data distribution and computational power in IoT devices.	Potential overhead from homomorphic encryption in fog computing environments.

might reduce the usefulness of the data when the distributions vary significantly.

3) **Computational Efficiency:** By distributing the computational workload across user devices, FL reduces the strain on centralized servers. This is particularly important in DCC, where participation can involve millions of devices, and relying on a central server would be impractical due to resource limitations and communication delays. Unlike MPC, which involves multiple rounds of secure exchanges that can be time-consuming and resource-intensive, FL conducts computations locally on devices. Only the aggregated model updates are sent to the server, which minimizes interaction and lowers the overall computational and communication costs. TEEs require specialized hardware for each participating device, making them impractical for large-scale DCC networks.
4) **Hardware Flexibility:** FL also offers greater flexibility in terms of hardware requirements compared with TEEs, which rely on specific hardware setups to secure data. In diverse DCC ecosystems, devices range widely in their capabilities from powerful servers to low-power IoT devices. The necessity for specialized hardware can complicate deployment and integration and limit scalability. However, FL adapts well to different types of devices, allowing organizations to leverage existing hardware without requiring special security components.
5) **Performance:** FL's incorporation of secure aggregation methods enhances its resilience against potential vulnerabilities such as side channels. Thus, FL is a balanced solution that maintains security while ensuring performance. Although MPC provides strong security through encrypted data sharing, its computation and communication overhead can lead to performance trade-offs that are unacceptable in real-time DCC scenarios. DP, while effectively preventing data leakage by adding noise, can reduce model accuracy, particularly in complex models that require minimal noise to function well. Similarly, although TEE methods offer robust security within their hardware environments, the need to consolidate data for secure processing can introduce inefficiencies that counteract the benefits of the DCC environment. In FL, adjustments can be made in real-time with minimal impact on the performance.

Overall, each generation of PPC evolves in response to the limitations or challenges posed by its predecessor, promoting the ongoing development of stronger and more efficient privacy-preserving solutions. Table 8 provides a summary of the four generations in terms of their security, performance, scalability, and efficiency.

AQ2: What challenges arise when using FL in DCC, and how can they be mitigated?

Utilizing FL in DCC has several challenges that must be addressed to ensure its effectiveness. One of the main issues is achieving convergence of the global model, which can be challenging owing to the diversity of data and the potential unreliability of updates from client devices. To tackle this problem, it is important to use adaptive learning rates and robust aggregation techniques such as Krum [170], Trimmed Mean [171], or FedShare [172]. These methods can help manage outliers and ensure the successful convergence of the model. In addition, the incorporation of regularization techniques can stabilize the training process.

Another challenge is the availability and reliability of the clients. Since clients may not always be available for training due to network issues or device limitations, this can disrupt the training process. A flexible participation strategy can help solve this problem, allowing clients to join and leave training as needed. Furthermore, addressing the issue of stragglers—clients who respond slowly—is crucial for maintaining the efficiency of the training. Resource constraints also play a significant role, as limited computational power and battery life on client devices can affect their ability to participate. To mitigate this, optimizing training algorithms for resource-constrained environments, such as Federated Transfer Learning [173], can enable clients to use pre-trained models.

In addition to these challenges, FL in DCC may face problems related to communication costs and network congestion. Transferring large model updates between clients and the cloud can slow the learning process, and frequent communication can lead to performance drops during peak usage times. As discussed in Section VI, a hierarchical approach to FL can be achieved by combining the strengths of both cloud and edge configurations. This structure is particularly beneficial for reducing communication across regions and ensuring that learning is accurate and relevant in specific areas.

Poisoning attacks, such as data poisoning and model poisoning, which can undermine the training process, are another critical concern in FL. Data poisoning involves altering input data to negatively affect outcomes, whereas model poisoning targets local model updates to introduce harmful behaviors into the global model [174], [175]. To combat such vulnerabilities, it is essential to use robust aggregation algorithms that are less sensitive to outliers along with leveraging anomaly detection and statistical information techniques. Establishing a reputation system for client whitelisting can further enhance security. It is also important to conduct validation checks on the data submitted by clients and implement model update validation to help identify and address potential poisoning vulnerabilities. Additionally, methods such as weight clipping [176] or weight watermarking [177] can limit the size of updates and unauthorized model tampering, preventing drastic changes from harmful inputs, while still allowing legitimate updates to be processed. Regularization techniques can also help

TABLE 8. Comparison of the four generations of PPC.

Generation	Security	Performance	Efficiency
MPC	Strong security as data is split among parties, preventing anyone from seeing the full dataset. It may be vulnerable to collision if parties combine their information; however, their security is supported by rigorous mathematical foundations.	Model performance is mostly preserved as computations occur on shared encrypted data, but some operations are limited or resource intensive. Not suited for complex AI models and computing functions.	Low efficiency owing to multiple rounds of secure communication and heavy computations, making it unsuitable for real-time systems or large datasets owing to high overhead.
DP	Privacy leakage still exists, especially with auxiliary data or multiple queries.	Model accuracy can be decreased, particularly with higher noise levels, requiring a balance between privacy and utility.	Suitable for small datasets, but computationally expensive for larger datasets or complex models owing to noise and query management.
TEE	Fortified security in hardware-protected enclaves, but vulnerable if the hardware is compromised.	Minimal effect on performance, as sensitive data operations occur in a secure environment, maintaining accuracy and utility.	Offers moderate efficiency owing to hardware-specific requirements and overhead from data exchange between secure and non-secure parts. However, a key drawback is the need to extract and consolidate the data from each party at a central TEE location.
FL	Robust security and privacy by keeping raw data on user devices and sharing only model updates. However, it remains susceptible to data poisoning, model poisoning, and malicious updates. Emerging methods, such as anomaly detection and data encryption, can help mitigate these vulnerabilities.	Strong performance levels comparable to those of centralized AI models.	High efficiency in distributed networks. However, large-scale updates may cause bandwidth and latency issues, with efficiency depending on communication channels between parties.

stabilize the training process and reduce the susceptibility to noisy updates.

AQ3: What are the advantages and disadvantages of each secure FL technique, including methods for detecting malicious updates, data encryption, data perturbation, and anonymization?

Each FL technique offers specific strengths and limitations, particularly in addressing the challenges within DCC systems. The ideal approach depends on the specific goals and resources available because each solution uniquely balances privacy, security, performance, accuracy, and efficiency. Table 9 shows a comparison of these techniques.

Data encryption is highly effective in maintaining data confidentiality during the entire computational and training process. This allows computations on encrypted data, meaning that sensitive information remains hidden from the view at all times. However, encryption has a significant cost that requires substantial computing power, especially when using advanced forms such as homomorphic encryption. Managing encryption keys in a DCC system can also be complex, which adds another layer of difficulty.

Data perturbation and additive masking provide an alternative by adding a level of noise to the gradient updates sent by clients, which reduces the required computing power compared with encryption. Carefully designed noise can preserve utility, allowing models to learn without exposing sensitive information, and this technique can be efficiently scaled across various model types. The main challenge with data perturbation is the balance between privacy and accuracy, which is similar to the issues encountered in DP. Higher privacy levels, achieved by adding more noise, often reduce the accuracy of a model. In addition, perturbation relies on a "privacy budget" to control how often data can be perturbed, thus limiting the number of updates that can be made before privacy becomes compromised.

Anonymization is attractive owing to its simplicity and low computational requirements. Removing identifiable information preserves data accuracy without altering the underlying data. However, anonymization is vulnerable to advanced attacks, particularly when adversaries have access to additional information. This makes it less reliable in situations where adversaries may attempt to re-identify individuals using supplementary data.

Anomaly detection strengthens FL by identifying and reacting to suspicious patterns or unusual data behaviors in real-time. This adaptability helps to prevent corrupted data from affecting the model. However, anomaly detection can sometimes misclassify data by flagging safe updates as malicious (false positives) or overlooking harmful updates (false negatives). Implementing this approach may also require additional computing power for constant monitoring, which may strain the resources.

Statistical information methods offer an efficient method to handle potentially malicious activities with relatively low computational demands. By tolerating some level of unusual behavior, this technique is robust against outliers and generally requires less ongoing maintenance, because it does not rely heavily on continuous learning. The primary drawback of these methods is that they are static and may struggle to recognize new vulnerability patterns over time. Additionally, aggressive filtering can reduce the model's accuracy.

TABLE 9. Advantages and disadvantages of privacy-preserving and secure FL techniques.

Technique	Pros	Cons
Data Encryption	- **Strong Security:** Ensures data remains confidential during computing and training. - **End-to-End Privacy:** Allows computations on encrypted data.	- **Computational Overhead:** Resource-intensive, especially with homomorphic encryption. - **Complex Key Management:** Challenging in DCC system.
Data Perturbation	- **Low Overhead:** Less intensive than encryption. - **Model Utility:** Carefully calibrated noise preserves data utility. - **Scalable:** Suitable for large datasets and various architectures.	- **Accuracy-Privacy Trade-off:** Higher privacy levels can reduce model accuracy. - **Privacy Budget Limitations:** Restricts the number of updates.
Anonymization	- **Simplicity:** Easy to implement and computationally efficient. - **Data Usability:** Preserves data accuracy since data is unaltered.	- **Weak Against Advanced vulnerabilities:** Can be compromised by linking auxiliary data.
Anomaly Detection	- **Real-Time Detection:** Identifies unusual patterns quickly. - **Adaptive:** Can learn and adapt to new malicious behaviors. - **Data Integrity:** Prevents corrupt data from influencing the model.	- **False Positive/Negative Rate:** May misclassify legitimate updates. - **Resource Intensive:** Requires additional computational resources for monitoring.
Statistical Information	- **Efficiency:** Lower computational requirements. - **Robustness:** Tolerates some malicious behavior, reducing outlier impact. - **Low Maintenance:** Minimal need for continuous learning.	- **Static Models:** Non-adaptive, may miss evolving vulnerable patterns. - **Accuracy Impact:** Aggressive filtering can affect model accuracy.
Blockchain	- **Transparency:** Immutable ledger provides accountability and traceability. - **Decentralized Trust:** Reduces reliance on a central authority. - **Strong Security:** Cryptographic foundations resist malicious activity.	- **Computational Overhead:** Consensus mechanisms add latency and increase computational costs.

Blockchain technology brings transparency to FL through an unchangeable record of activities that promotes accountability and traceability. Its decentralized nature removes the need for a central authority, reducing the risks of single points of failure, whereas its cryptographic foundations make it highly resistant to malicious interference. However, consensus mechanisms that maintain blockchain integrity are computationally intensive and can slow down DCC systems.

AQ4: How can FL establish a fair balance between security, performance, efficiency, and accuracy in DCC?

Hierarchical FL effectively balances security, privacy, performance, accuracy, and efficiency in DCC by combining a decentralized framework with the benefits of both cloud and edge configurations, along with adaptive security measures. Sensitive data remain on user devices, thereby reducing the risk of exposing the raw information. Additional protections, such as data perturbation and anonymization, render FL more resistant to data breaches. Furthermore, FL's secure aggregation methods maintain the accuracy of the model while safeguarding user data. With added features, such as leveraging statistical information, anomaly detection, and robust aggregation algorithms, FL can address data poisoning, model tampering, and malicious updates [178], [179]. Although data encryption and blockchain significantly enhance security guarantees, their computational load can disrupt balance. Therefore, they are recommended only for cloud environments that require a very high level of privacy and security.

On the performance side, FL distributes computing tasks across millions of devices instead of relying on centralized servers. Local processing on each device helps avoid communication slowdowns, enabling FL to perform well even in environments with limited bandwidth or high traffic. The local training of models on user data also enables FL to capture unique location-specific patterns, thereby strengthening the predictive capability of the global model. By transmitting only model updates to the central server rather than whole datasets, FL lowers latency and optimizes data transfer, which is particularly valuable in DCC settings, where quick responses are critical. Methods such as Federated Averaging improve model performance by allowing FL to manage diverse data more effectively than other privacy-preserving techniques such as MPC or TEE. FL is highly flexible with hardware, and seamlessly operates across a variety of devices, from robust edge servers to simpler IoT devices, making it easy to integrate into current DCC systems while minimizing setup costs.

VIII. OPEN RESEARCH DIRECTIONS

Future research on DCC, particularly in terms of privacy and security, offers numerous directions for improving FL and other PPC generations. Here, we discuss key areas for future work, including ways to enhance security, optimize performance, and develop user-centered solutions. Table 10 provides a summary of open research areas along with their key directions.

Optimizing Security, Performance, and Efficiency in MPC, DP, and TEE: MPC allows secure calculations but is susceptible to privacy vulnerabilities when participants collude. Adaptive protocols that are resistant to collusion can be developed to detect and counteract such collusions using the probabilistic behavior of the participants. Integrating these protocols with DP can further protect the

aggregated data analysis. However, DP also faces vulnerabilities, particularly when adversaries exploit auxiliary data to infer private information. Solutions, such as adaptive noise mechanisms and Bayesian DP models [180], can strengthen DP resilience. Thus, future research should explore advanced privacy-preserving noise injection strategies. TEEs provide additional security by isolating sensitive data, although hardware dependency often limits it. Research should focus on the development of software-defined or virtual TEEs and lightweight secure enclave architectures that reduce reliance on proprietary hardware. MPC's high computational and communication demands necessitate optimizing cryptographic algorithms. Additionally, addressing the challenges of data transfer in PPC may involve the creation of efficient communication protocols (e.g., low-latency encrypted channels and proactive caching mechanisms), hierarchical DP techniques, and virtual distributed TEEs to improve scalability and reduce latency across secure applications.

Optimizing Privacy-Preserving and Secure FL Techniques: Current privacy-preserving and secure techniques in FL, such as data encryption and blockchain, are essential but still have substantial computational and processing demands. Traditional encryption methods require significant power and time, which hinders their scalability in settings with limited resources such as mobile and edge devices. Research is increasingly focused on developing homomorphic encryption, which allows computations of encrypted data without decryption. Although promising for privacy, homomorphic encryption itself is resource-intensive. Therefore, future work should develop hybrid encryption models to balance security and efficiency and make it suitable for real-time FL applications. Extreme learning machine offers fast and efficient learning for FL by eliminating iterative weight tuning, making it ideal for resource-constrained environments like edge computing. However, its security against adversarial attacks and privacy risks remains underexplored. Future research should integrate extreme Learning with privacy-preserving methods to enhance security while maintaining efficiency [181], [182]. Deep active learning also improves FL by selecting key data samples for labeling, reducing annotation costs and computation. It accelerates learning but raises privacy concerns in decentralized settings. Optimizing secure deep active methods for low-bandwidth environments remains a key research area [183]. Additionally, FL-blockchain consensus mechanisms are being explored to secure operations across distributed clouds. Standard Proof of Work protocols are not well-suited for FL due to high energy consumption and latency [184]. Instead, lightweight alternatives, such as IOTA and HashGraph, offer more efficient solutions, reducing processing overhead while maintaining high transaction throughput and security with minimal resource use [185]. More research is needed to integrate directed acyclic graph-based consensus mechanisms with FL frameworks to achieve scalable, privacy-aware coordination among distributed nodes.

Adversarial Training and Self-Healing Mechanisms: FL systems are susceptible to adversarial vulnerabilities, particularly data poisoning, where malicious actors manipulate training data to degrade performance [186], [187]. Adversarial training like GANs can counteract these vulnerabilities by exposing models to synthetic adversarial examples during training, thereby helping them learn to identify and resist manipulative patterns. However, GAN-based approaches can introduce mode collapse; therefore, novel meta-learning adversarial training strategies should be investigated to enhance model generalization. Additionally, dynamic algorithms can enhance the FL model adaptability by learning from and adjusting to known vulnerable signatures. Developing federated reinforcement learning frameworks that adaptively update security policies in response to detected adversarial behavior will further strengthen FL security. Self-healing mechanisms [188], such as periodic snapshots and rollback protocols, can add another layer of resilience by allowing systems to revert to a secure state following an attack, automating the detection and repair of compromised model components to maintain integrity without human intervention.

Game-Theoretic Strategies: Game theory models the interactions between adversaries and defenders, and provides a framework for predicting and countering potential vulnerabilities in privacy systems. In FL and DCC, game-theoretic approaches such as Stackelberg games can allow defenders to develop preemptive strategies, positioning themselves advantageously before an attack occurs [189]. Thus, future research should explore stochastic game-theoretic frameworks that consider uncertainties in adversary actions to design more robust countermeasures.

Multi-Layer User-Centric Privacy Solutions: User-centered privacy is essential in DCC because users have diverse preferences regarding data privacy and participation levels. Privacy solutions tailored to individual preferences allow users to choose privacy settings based on data sensitivity, thereby giving them more control. This can involve an adjustable privacy slider or other intuitive interfaces that allow users to manage their level of involvement. Elements of gamification, such as privacy-related rewards, could incentivize users to actively manage their settings, thus encouraging a privacy-conscious data-sharing culture. Multi-layered privacy controls could provide users with granular privacy options tailored to each layer of application interaction, from data entry to processing and storage. A priority research area involves designing automated privacy-adaptive systems that adjust permissions dynamically based on contextual factors such as location, data type, and past user behaviors. These multi-tenant environments can benefit from real-time compliance monitoring and privacy assessments to ensure that shared platforms dynamically meet privacy standards across various users and applications.

Real-World Applications and Case Studies: Testing FL and other PPC techniques in real-world applications is essential to assess their effectiveness. Sectors such as health-

care, where patient confidentiality is critical, and smart city infrastructure, provide rich environments for practical trials. Privacy-preserving analytics in electronic health records, secure vehicular data sharing in intelligent transportation systems, and confidential transaction verification in financial services represent key domains requiring further research. Case studies could reveal insights into the operational benefits of FL, evaluate privacy and security impacts, establish benchmarks for PPC effectiveness, and guide future research and implementation.

Cross-Layer Privacy Mechanisms: Cross-layer privacy integration in DCC can secure data throughout the system, from application to infrastructure layers, creating a comprehensive defense against a range of security challenges. Developing privacy-aware middleware that ensures consistent data protection across different cloud layers through unified encryption policies and adaptive access control mechanisms is a critical area of research.

Quantum Security: With the rise in quantum computing, traditional cryptographic methods are at risk, making quantum-resistant security a top priority. Developing efficient and scalable post-quantum cryptographic algorithms is essential for protecting DCC and FL systems. Another key challenge is quantum key distribution, which offers strong security but is difficult to implement on a large scale. Further research should focus on integrating this with existing networks to make it more practical. In addition, secure key exchange mechanisms must be redesigned to remain effective against quantum threats without adding excessive complexity. Beyond encryption, quantum machine learning has the potential to transform cybersecurity by detecting threats; however, more work is needed to make it applicable in real-world scenarios. To ensure a smooth transition to quantum-resistant security, standardization and interoperability must be prioritized.

Long-Term Studies on Privacy Systems: Privacy systems must adapt to changes in user preferences and emerging vulnerabilities. Longitudinal studies can provide valuable insights into user trust and behavioral patterns over time, allowing privacy frameworks to evolve in response to users' needs. Researchers can refine privacy metrics and techniques by studying privacy systems across diverse demographics and developing adaptive privacy mechanisms to ensure their ongoing relevance and effectiveness in real-world conditions.

Ethics and Regulation in Privacy: Privacy frameworks must comply with data protection laws to remain viable. Research on compliance-by-design architectures can ensure that privacy protocols adapt to regulatory shifts, creating systems that inherently align with current and future laws. Frameworks that emphasize fairness and transparency are crucial for preventing biased outcomes and promoting user trust. In addition, research on consent mechanisms and trust-building methods can enhance user confidence in privacy protocols and ensure that systems align with shifting user expectations. A promising direction is the development of regulatory sandboxes where privacy-preserving techniques can be experimentally validated under real-world regulatory constraints before full-scale deployment.

Implementing Zero-Knowledge Proofs: Zero-knowledge proofs provide a powerful tool that enables verification of data integrity without revealing sensitive information [190]. This allows participants to prove data authenticity or computational correctness without disclosing data, which is invaluable for the decentralized nature of DCC. Lightweight zero-knowledge protocols could optimize their application, making them suitable for resource-constrained environments. This research area focuses on creating efficient proof systems that can securely validate computations across untrusted networks. Future research should focus on reducing proof generation times and optimizing proof verification processes to enhance scalability in real-time cloud environments.

Maintaining Service Quality in High-Mobility DCC: In high-mobility DCC environments, users frequently change network zones, thereby creating challenges for data continuity and service quality. Research should focus on service migration strategies and predictive models that track user mobility to ensure a seamless transition. Machine learning-driven mobility prediction algorithms can help anticipate user transitions and proactively reallocate computing resources. Caching strategies can temporarily store data to improve access, while the adaptive Quality of Service frameworks can adjust to varying network conditions. Additionally, efficient handover mechanisms could reduce delays, helping maintain consistent data access in dynamic mobile DCC environments. Future research should explore integrating edge intelligence with federated caching to dynamically adapt storage and computation based on real-time mobility patterns.

IX. CONCLUSION

The survey highlighted a significant shift from centralized cloud models to DCC environments, driven by the increasing number of connected devices and the rise of data-intensive applications. It identified unique security and privacy concerns associated with DCC and provided a comprehensive classification of PPC generations designed to address these challenges. In each category, the review examined the most significant research conducted over the past decade in various prestigious conferences and journals, analyzing the methods used, the challenges addressed, and strengths and weaknesses of each approach. Particular attention is given to FL, as the research concluded that it holds greater potential than other generations, such as MPC, DP, and TEE, in tackling the privacy and security issues faced by DCC. The decentralized nature of FL aligns well with that of DCC, offering several advantages: it can handle diverse data, achieve strong performance and high efficiency in distributed systems, and adapt to different hardware configurations. The review also identified the challenges that FL faces in DCC, including the convergence of the global model, management

TABLE 10. Summary of Open Research Directions

Research Area	Key Directions	Research Area	Key Directions
MPC, DP, and TEE Optimization	- Adaptive collusion-resistant MPC. - Adaptive noise and Bayesian DP models. - Virtual TEEs and lightweight enclaves. - Low-latency encrypted channels and proactive caching mechanisms.	**Game-Theoretic Strategies**	- Stochastic game-theoretic models. - Stackelberg game for defense.
Optimizing FL Techniques	- Hybrid encryption models. - Secure extreme learning in FL. - Deep active learning for low-bandwidth. - FL-blockchain with directed acyclic graph-based consensus mechanisms.	**User-Centric Privacy**	- Adjustable privacy sliders. - Multi-layer privacy controls and real-time compliance monitoring. - Context-based privacy adaptation. - Gamification techniques.
Adversarial Training and Self-Healing	- GAN-based adversarial training. - Meta-learning adversarial training strategies. - Federated reinforcement learning security. - Self-healing rollback protocols.	**Real-World Applications**	- Privacy-preserving healthcare analytics. - Secure vehicular data sharing. - Confidential transactions in finance.
Cross-Layer Privacy	- Middleware for cross-layer security.	**Quantum Security**	- Post-quantum cryptography research. - Optimized quantum-resistant key exchange. - Standardization and interoperability.
Long-Term Privacy Studies	- Longitudinal user behavior analysis. - Privacy evaluations across demographics.	**Ethics and Regulation**	- Compliance-by-design architectures. - Fair and transparent privacy frameworks. - Regulatory sandboxes for experimental validation.
Zero-Knowledge Proofs	- Lightweight zero-knowledge proofs for low-power devices. - Optimization of proof verification.	**Maintaining Service Quality in DCC**	- ML-driven mobility prediction. - Federated caching for adaptation.

of unreliable client devices, concerns about communication costs and network congestion, and certain vulnerabilities, such as data and model poisoning. To address these issues, this review discusses various solution techniques, including data encryption and perturbation, robust aggregation, flexible client engagement, federated transfer learning, blockchain technology, anomaly detection, and leveraging statistical information. A thorough comparison of these FL techniques is presented, highlighting their respective advantages and disadvantages. Finally, future research directions are introduced to assist researchers and policymakers in the development of effective security strategies for real-world DCC applications.

X. Acknowledge

This work was conducted as part of the SUSTAINET-Advance (16KIS2280) and 6G-RIC (16KISK032) projects.

REFERENCES

[1] X. Q. Pham, T. D. Nguyen, T. H. Huynh-The, E. Huh, and D. S. Kim, "Distributed cloud computing: architecture, enabling technologies, and open challenges," *IEEE Consumer Electronics Magazine*, vol. 12, no. 3, pp. 98–106, 2022.

[2] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, and M. Shoaib, "Bringing computation closer toward the user network: Is edge computing the solution?" *IEEE Communications Magazine*, vol. 55, no. 11, pp. 138–144, 2017.

[3] A. J. Ferrer, J. M. Marquès, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–36, 2019.

[4] Q. Duan, S. Wang, and N. Ansari, "Convergence of networking and cloud/edge computing: Status, challenges, and opportunities," *IEEE Network*, vol. 34, no. 6, pp. 148–155, 2020.

[5] S. A. Bello, L. O. Oyedele, O. O. Akinade, M. Bilal, J. M. D. Delgado, L. A. Akanbi, A. O. Ajayi and H. A. Owolabi, "Cloud computing in construction industry: Use cases, benefits and challenges," *Automation in Construction*, vol. 122, 103441, 2021.

[6] A. K. Sandhu, "big data with cloud computing: discussions and challenges," *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32–40, 2021.

[7] F. Arena, M. Collotta, G. Pau, and F. Termine, "An overview of augmented reality," *Computers*, vol. 11, no. 2, article no. 28, 2022.

[8] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[9] Q. Yang, "Toward Responsible AI: An Overview of Federated Learning for User-centered Privacy-preserving Computing," *ACM Transactions on Interactive Intelligent Systems*, vol. 11, no. 3–4, pp. 1–22, 2021.

[10] X. Yin, Y. Zhu, and J. Hu, "A Comprehensive Survey of Privacy-preserving Federated Learning: A Taxonomy, Review, and Future Directions," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–36, 2021.

[11] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, 2021.

[12] H. G. Abreha, M. Hayajneh, and M. A. Serhani, "Federated Learning in Edge Computing: A Systematic Survey," *Sensors*, vol. 22, no. 2, 2022.

[13] A. Brecko, E. Kajati, J. Koziorek, and I. Zolotova, "Federated learning for edge computing: A survey," *Applied Sciences*, vol. 12, no. 18, p. 9124, 2022.

[14] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, vol. 10, no. 2, pp. 194–213, 2024.

[15] J. Chen, H. Yan, Z. Liu, M. Zhang, H. Xiong, and S. Yu, "When federated learning meets privacy-preserving computation," *ACM Computing Surveys*, vol. 56, no. 12, pp. 1–36, 2024.

[16] H. Li, L. Ge, and L. Tian, "Survey: Federated learning data security and privacy-preserving in edge-Internet of Things," *Artificial Intelligence Review*, vol. 57, no. 5, p. 130, 2024.

[17] Q. Han, S. Lu, W. Wang, H. Qu, I. Li, and Y. Gao, "Privacy-preserving and secure robust federated learning: A survey," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 13, 2024.

[18] S. Saha, A. Hota, A. K. Chattopadhyay, A. Nag, and S. Nandi, "A multifaceted survey on privacy preservation of federated learning:

progress, challenges, and opportunities," *Artificial Intelligence Review*, vol. 57, no. 7, 2024.

[19] Y. Coady, O. Hohlfeld, J. Kempf, R. McGeer, and S. Schmid, "Distributed cloud computing: Applications, status quo, and challenges," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 2, pp. 38–43, 2015.

[20] P. Khethavath, J. P. Thomas, and E. Chan-Tin, "Towards an efficient distributed cloud computing architecture," *Peer-to-Peer Networking and Applications*, vol. 10, pp. 1152–1168, 2017.

[21] B. O. Akram, N. K. Noordin, F. Hashim, M. F. A. Rasid, M. I. Salman, and A. M. Abdulghani, "Enhancing Reliability of Time-Triggered Traffic in Joint Scheduling and Routing Optimization within Time-Sensitive Networks," *IEEE Access*, 2024.

[22] A. M. Abdulghani, A. Abdullah, A. R. Rahiman, N. A. W. A. Hamid, B. O. Akram, and H. Raissouli, "Navigating the Complexities of Controller Placement in SD-WANs: A Multi-Objective Perspective on Current Trends and Future Challenges," *Computer Systems Science & Engineering*, vol. 49, 2025.

[23] A. M. Abdulghani, "Hybrid task scheduling algorithm for makespan optimisation in cloud computing: A performance evaluation," *J. Artif. Intell.*, vol. 6, no. 1, pp. 241-259, 2024.

[24] H. Tabrizchi and M. K. Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," *The Journal of Supercomputing*, vol. 76, no. 12, pp. 9493–9532, 2020.

[25] J. S. Ng, W. Y. B. Lim, N. C. Luong, Z. Xiong, A. Asheralieva, D. Niyato, and C. Miao, "A comprehensive survey on coded distributed computing: Fundamentals, challenges, and networking applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1800–1837, 2021.

[26] S. H. H. Madni, M. S. Abd Latiff, and Y. Coulibaly, "Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities," *Journal of Network and Computer Applications*, vol. 68, pp. 173–200, 2016.

[27] C. Pahl, "Containerization and the PaaS cloud," *IEEE Cloud Computing*, vol. 2, no. 3, pp. 24–31, 2015.

[28] E. Van Eyk, L. Toader, S. Talluri, L. Versluis, A. Uță, and A. Iosup, "Serverless is more: From PaaS to present cloud computing," *IEEE Internet Computing*, vol. 22, no. 5, pp. 8–17, 2018.

[29] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A survey on resiliency techniques in cloud computing infrastructures and applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2244–2281, 2016.

[30] D. Firestone, A. Putnam, S. Mundkur *et al.*, "Azure accelerated networking: SmartNICs in the public cloud," *In 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pp. 51-66, 2018.

[31] J. Surbiryala and C. Rong, "Cloud Computing: History and Overview," in *IEEE Cloud Summit*, Washington, DC, USA, 2019.

[32] K. Wang, C. Zhao, J. Chu, Y. Shi, J. Lu, B. Lyu, S. Zhu, P. Cheng, and J. Chen, "LFVeri: Network Configuration Verification for Virtual Private Cloud Networks," *IEEE/ACM Transactions on Networking,* 2024.

[33] A. Gordon, "The Hybrid Cloud Security Professional," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 82–86, 2016.

[34] I. Petri, J. Diaz-Montes, O. Rana, M. Punceva, I. Rodero, and M. Parashar, "Modelling and implementing social community clouds," *IEEE Transactions on Services Computing*, vol. 10, no. 3, pp. 410–422, 2015.

[35] E. Levin, N. Beisekenov, M. Wilson, M. Sadenova, R. Nabaweesi, and L. Nguyen, "Empowering climate resilience: Leveraging cloud computing and big data for community Climate Change Impact Service (C3IS)," *Remote Sensing,* vol. 15, no. 21, p. 5160, 2023.

[36] Y. Zhou, B. Yang, H. Hou, L. Zhang, T. Wang, and M. Hu, "Continuous leakage-resilient identity-based encryption with tight security," *The Computer Journal*, vol. 62, no. 8, pp. 1092–1105, 2019.

[37] F. Castro-Medina, L. Rodriguez-Mazahua, A. López-Chau, M. A. Abud-Figueroa, and G. Alor-Hernández, "FRAGMENT: A web application for database fragmentation, allocation and replication over a cloud environment," *IEEE Latin America Transactions*, vol. 18, no. 6, pp. 1126–1134, 2020.

[38] H. Pan, Z. Li, P. Zhang, P. Cui, K. Salamatian, and G. Xie, "Misconfiguration-free compositional SDN for cloud networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2484–2499, 2022.

[39] H. Wang, X. Yi, E. Bertino, and L. Sun, "Protecting outsourced data in cloud computing through access management," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 3, pp. 600–615, 2016.

[40] Y. Sharma, B. Javadi, W. Si, and D. Sun, "Reliability and energy efficiency in cloud computing systems: Survey and taxonomy," *Journal of Network and Computer Applications*, vol. 74, pp. 66–85, 2016.

[41] X. Gong, Y. Chen, Q. Wang, H. Huang, L. Meng, C. Shen, and Q. Zhang, "Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2617–2631, 2021.

[42] L. Hou, S. Zhao, X. Li, P. Chatzimisios, and K. Zheng, "Design and implementation of application programming interface for Internet of things cloud," *International Journal of Network Management*, vol. 27, no. 3, p. e1936, 2017.

[43] A. Nag, M. M. Hassan, A. Das, A. Sinha, N. Chand, A. Kar, V. Sharma, and A. Alkhayyat, "Exploring the applications and security threats of Internet of Things in the cloud computing paradigm: A comprehensive study on the cloud of things," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 4, p. e4897, 2024.

[44] S. Rizvi and I. Williams, "Analyzing Transparency and Malicious Insiders Prevention for Cloud Computing Environment," *Computers & Security*, vol. 137, p. 103622, 2024.

[45] M. Kim, H. Yang, and J. Lee, "Fully private coded matrix multiplication from colluding workers," *IEEE Communications Letters*, vol. 25, no. 3, pp. 730–733, 2020.

[46] Y. Ma, C. Yu, and C. Weng, "Morpheus: An efficient timing-based attestation framework for safeguarding hypervisor integrity with dynamic trust," *Computers & Security*, vol. 144, p. 103966, 2024.

[47] L. Chhaya, P. Sharma, G. Bhagwatikar, and A. Kumar, "Wireless Sensor Network Based Smart Grid Communications: Cyber Attacks, Intrusion Detection System and Topology Control," *Electronics*, vol. 6, no. 1, 2017.

[48] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in IoT: a survey," *The Journal of Supercomputing*, vol. 76, pp. 5320–5363, 2020.

[49] J. Zhang, C. Chen, J. Cui, and K. Li, "Timing Side-Channel Attacks and Countermeasures in CPU Microarchitectures," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1-40, 2024.

[50] M. A. Petcu, M. I. Sobolevschi-David, and S. C. Curea, "Integrating Digital Technologies in Sustainability Accounting and Reporting: Perceptions of Professional Cloud Computing Users," *Electronics*, vol. 13, no. 14, p. 2684, 2024.

[51] C. Zuo, Z. Lin, and Y. Zhang, "Why does your data leak? Uncovering the data leakage in cloud from mobile apps," *IEEE Symposium on Security and Privacy (SP)*, pp. 1296–1310, 2019.

[52] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way," *Journal of Network and Computer Applications*, vol. 84, pp. 38–54, 2017.

[53] A. Albugmi, M. O. Alassafi, R. Walters, and G. Wills, "Data security in cloud computing," in *Fifth International Conference on Future Generation Communication Technologies (FGCT)*, London, UK, 2016.

[54] A. Ahmed, R. Latif, S. Latif, H. Abbas, and F. A. Khan, "Malicious insiders attack in IoT based multi-cloud e-healthcare environment: a systematic literature review," *Multimedia Tools and Applications*, vol. 77, pp. 21947–21965, 2018.

[55] L. L. Dhirani, N. Mukhtiar, B. S. Chowdhry, and T. Newe, "Ethical dilemmas and privacy issues in emerging technologies: A review," *Sensors*, vol. 23, no. 3, p. 1151, 2023.

[56] P. Goswami, N. Faujdar, S. Debnath, A. K. Khan, and G. Singh, "Investigation on Storage Level Data Integrity Strategies in Cloud Computing: Classification, Security Obstructions, Challenges and Vulnerability," *Journal of Cloud Computing*, vol. 13, no. 45, 2024.

[57] A. Tundo, M. Mobilio, O. Riganelli, and L. Mariani, "Monitoring Probe Deployment Patterns for Cloud-Native Applications: Definition and Empirical Assessment," *IEEE Transactions on Services Computing*, vol. 17, no. 4, pp. 1636–1654, 2024.

[58] C. Li, Z. Qin, E. Novak, and Q. Li, "Securing SDN Infrastructure of IoT–Fog Networks from MitM Attacks," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1156–1164, 2017.

[59] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," *Frontiers in Computer Science*, vol. 3, 2021.

[60] C. Zhang, Z. DeStefano, A. Arun, J. Bonneau, P. Grubbs, and M. Walfish, "Zombie: Middleboxes that Don't Snoop," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pp. 1917–1936, 2024.

[61] A. Rahdari, A. Jalili, M. Esnaashari, M. Gheisari, A. A. Vorobeva, Z. Fang, *et al.*, "Security and Privacy Challenges in SDN-Enabled IoT Systems: Causes, Proposed Solutions, and Future Directions," *Computers, Materials & Continua,* vol. 80, no. 2, pp. 2511–2533, 2024.

[62] M. Sookhak, A. Gani, H. Talebian, A. Akhunzada, S. U. Khan, R. Buyya, and A. Y. Zomaya, "Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1–34, 2015.

[63] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Computer Science Review*, vol. 33, pp. 1–48, 2019.

[64] A. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pp. 160–164, 1982.

[65] A. C. C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (SFCS 1986)*, pp. 162–167, 1986.

[66] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[67] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-Functional Distributed Computing," in *IEEE 60th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1–8, 2024.

[68] A. Khalesi, M. Mirmohseni, and M. A. Maddah-Ali, "The capacity region of distributed multi-user secret sharing," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 1057–1071, 2021.

[69] J. B. Almeida, M. Barbosa, G. Barthe, H. Pacheco, V. Pereira, and B. Portela, "Enforcing ideal-world leakage bounds in real-world secret sharing MPC frameworks," in *IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 132–146, July 2018.

[70] M. Yung, "From mental poker to core business: Why and how to deploy secure computation protocols?," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1–2, Oct. 2015.

[71] A. Khalesi, S. Daei, M. Kountouris, and P. Elia, "Multi-user distributed computing via compressed sensing," in *IEEE Information Theory Workshop (ITW)*, pp. 509–514, 2023.

[72] C. Hong, J. Katz, V. Kolesnikov, W.-j. Lu, and X. Wang, "Covert security with public verifiability: Faster, leaner, and simpler," in *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 38, pp. 97–121, May 2019.

[73] K. Eldefrawy, S. Hwang, R. Ostrovsky, and M. Yung, "Communication-efficient (proactive) secure computation for dynamic general adversary structures and dynamic groups," in *Security and Cryptography for Networks: 12th International Conference, SCN 2020*, Amalfi, Italy, pp. 3–22, September 2020.

[74] P. Mohassel, M. Rosulek, and Y. Zhang, "Fast and secure three-party computation: The garbled circuit approach," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2015, pp. 591–602.

[75] S. S. Burra, E. Larraia, J. B. Nielsen, P. S. Nordholt, C. Orlandi, E. Orsini, P. Scholl, and N. P. Smart, "High-performance multi-party computation for binary circuits based on oblivious transfer," *Journal of Cryptology*, vol. 34, no. 3, p. 34, 2021.

[76] N. Volgushev, M. Schwarzkopf, B. Getchell, et al., "Conclave: Secure multi-party computation on big data," in *Proceedings of the Fourteenth EuroSys Conference 2019*, pp. 1–18, Association for Computing Machinery, 2019.

[77] R. Cartor, R. G. D'Oliveira, S. E. Rouayheb, D. Heinlein, D. Karpuk, and A. Sprintson, "Secure distributed matrix multiplication with pre-computation," in *2024 IEEE International Symposium on Information Theory (ISIT)*, Jul. 2024, pp. 2568–2573.

[78] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[79] X. Ma, F. Zhang, X. Chen, and J. Shen, "Privacy preserving multi-party computation delegation for deep learning in cloud computing," *Information Sciences*, vol. 459, pp. 103–116, 2018.

[80] C. Gao and J. Yu, "SecureRC: A system for privacy-preserving relation classification using secure multi-party computation," *Computers & Security*, vol. 128, 2023.

[81] H. Akbari-Nodehi and M. A. Maddah-Ali, "Secure Coded Multi-Party Computation for Massive Matrix Operations," *IEEE Transactions on Information Theory*, vol. 67, no. 4, pp. 2379–2398, 2021.

[82] H. Akbari Nodehi and M. A. Maddah-Ali, "Limited-Sharing Multi-Party Computation for Massive Matrix Operations," in *IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, 2018.

[83] H. Akbari Nodehi, S. R. Hoseini Najarkolaei, and M. A. Maddah-Ali, "Entangled Polynomial Coding in Limited-Sharing Multi-Party Computation," in *IEEE Information Theory Workshop (ITW)*, Guangzhou, China, 2018.

[84] X. Lu, U. Y. Basaran, and B. Güler, "Scalable Multi-Round Multi-Party Privacy-Preserving Neural Network Training," *IEEE Transactions on Information Theory*, 2024.

[85] A. Khalesi and P. Elia, "Multi-user linearly separable computation: A coding theoretic approach," in *2022 IEEE Information Theory Workshop (ITW)*, pp. 428–433, Nov. 2022.

[86] A. Khalesi and P. Elia, "Multi-user linearly-separable distributed computing," *IEEE Transactions on Information Theory*, vol. 69, no. 10, pp. 6314–6339, 2023.

[87] L. Ma, B. Duan, B. Zhang, Y. Li, Y. Fu, and D. Ma, "A trusted IoT data sharing method based on secure multi-party computation," *Journal of Cloud Computing*, vol. 13, no. 1, p. 138, 2024.

[88] H. Kaur, N. Kumar, and S. Batra, "ClaMPP: A cloud-based multi-party privacy preserving classification scheme for distributed applications," *The Journal of Supercomputing*, vol. 75, pp. 3046–3075, 2019.

[89] V. Sucasas, A. Aly, G. Mantas, J. Rodriguez, and N. Aaraj, "Secure multi-party computation-based privacy-preserving authentication for smart cities," *IEEE Transactions on Cloud Computing*, vol. 11, no. 4, pp. 3555–3572, 2023.

[90] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-Server Multi-Function Distributed Computation," *Entropy*, vol. 26, no. 6, pp. 448, 2024.

[91] J. Laeuchli, Y. Ramírez-Cruz, and R. Trujillo-Rasua, "Analysis of centrality measures under differential privacy models," *Applied Mathematics and Computation*, vol. 412, p. 126546, 2022.

[92] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Proceedings of the 3rd Theory of Cryptography Conference*, pp. 265–284, 2006.

[93] C. Dwork, "Differential privacy," *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)(2)*, pp. 1–12, 2006.

[94] Y. Zhao and J. Chen, "A survey on differential privacy for unstructured data content," *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1–28, 2022.

[95] X. Xiong, S. Liu, D. Li, Z. Cai, and X. Niu, "A comprehensive survey on local differential privacy," *Security and Communication Networks*, vol. 2021, no. 1, 2021.

[96] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and D. Megías, "Individual differential privacy: A utility-preserving formulation of differential privacy guarantees," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1418–1429, 2017.

[97] Y. X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled Rényi differential privacy and analytical moments accountant," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (PMLR)*, pp. 1226–1235, Apr. 2019.

[98] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[99] X. Lyu, "Composition theorems for interactive differential privacy," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 9700–9712, 2022.

[100] K. Zhu, P. Van Hentenryck, and F. Fioretto, "Bias and variance of post-processing in differential privacy," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 11177–11184, May 2021.

[101] H. Jiang, J. Pei, D. Yu, J. Yu, B. Gong, and X. Cheng, "Applications of differential privacy in social network analysis: A survey," *IEEE*

Transactions on Knowledge and Data Engineering, vol. 35, no. 1, pp. 108–127, 2021.

[102] T. Wang, Y. Mei, W. Jia, X. Zheng, G. Wang, and M. Xie, "Edge-based differential privacy computing for sensor–cloud systems," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 75–85, 2020.

[103] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang, "A differential privacy-based query model for sustainable fog data centers," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 2, pp. 145–155, 2017.

[104] W. Fan, J. He, M. Guo, P. Li, Z. Han, and R. Wang, "Privacy preserving classification on local differential privacy in data centers," *Journal of Parallel and Distributed Computing*, vol. 135, pp. 70–82, 2020.

[105] K. Gai, Y. Wu, L. Zhu, Z. Zhang, and M. Qiu, "Differential privacy-based blockchain for industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4156–4165, 2019.

[106] C. Piao, Y. Shi, J. Yan, C. Zhang, and L. Liu, "Privacy-preserving governmental data publishing: A fog-computing-based differential privacy approach," *Future Generation Computer Systems*, vol. 90, pp. 158–174, 2019.

[107] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5827–5842, 2019.

[108] Z. Lv and F. Piccialli, "The Security of Medical Data on Internet Based on Differential Privacy Technology," *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–18, 2021.

[109] A. Ganesh, M. Haghifam, T. Steinke, and A. G. Thakurta, "Faster Differentially Private Convex Optimization via Second-Order Methods," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[110] K. Amin, A. Kulesza, A. Munoz, and S. Vassilvtiskii, "Bounding User Contributions: A Bias-Variance Trade-off in Differential Privacy," in *Proceedings of the 36th International Conference on Machine Learning (PMLR)*, PP. 263–271, Long Beach, California, USA, 2019.

[111] R. Chourasia, J. Ye, and R. Shokri, "Differential Privacy Dynamics of Langevin Diffusion and Noisy Gradient Descent," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 14771–14781, 2021.

[112] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palanis, "PPFA: Privacy Preserving Fog-Enabled Aggregation in Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018.

[113] M. Bi, Y. Wang, Z. Cai, and X. Tong, "A privacy-preserving mechanism based on local differential privacy in edge computing," *China Communications*, vol. 17, no. 9, pp. 50–65, 2020.

[114] P. Zhao, Z. Yang, and G. Zhang, "Personalized and differential privacy-aware video stream offloading in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 12, no. 1, pp. 347–358, 2024.

[115] S. D. Okegbile, J. Cai, H. Zheng, J. Chen, and C. Yi, "Differentially Private Federated Multi-Task Learning Framework for Enhancing Human-to-Virtual Connectivity in Human Digital Twin," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3533–3547, 2023.

[116] W. Jung, S. Kwon, and K. Shim, "TIDY: Publishing a Time Interval Dataset with Differential Privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 2280–2294, 2021.

[117] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," *2015 IEEE Trustcom/BigDataSE/Ispa*, vol. 1, pp. 57–64, Aug. 2015.

[118] P. Jauernig, A. R. Sadeghi, and E. Stapf, "Trusted execution environments: Properties, applications, and challenges," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 56–60, 2020.

[119] Open Mobile Terminal Platform, "Advanced Trusted Environment: OMTP TR1," 2009. [Online]. Available: http://omtp.org/OMTP_Advanced_Trusted_Environment_OMTP_TR1_v1_1.pdf. Accessed on: May 28, 2009.

[120] A. Muñoz, R. Rios, R. Román, and J. López, "A survey on the (in)security of trusted execution environments," *Computers & Security*, vol. 129, p. 103180, 2023.

[121] C. Shepherd, G. Arfaoui, I. Gurulian, R. P. Lee, K. Markantonakis, R. N. Akram, D. Sauveron, and E. Conchon, "Secure and trusted execution: Past, present, and future–A critical review in the context of the internet of things and cyber-physical systems," *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 168–177, 2016.

[122] S. Pinto and N. Santos, "Demystifying ARM TrustZone: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[123] Y. Fan, S. Liu, G. Tan, and F. Qiao, "Fine-grained access control based on Trusted Execution Environment," *Future Generation Computer Systems*, vol. 109, pp. 551–561, 2020.

[124] D. C. G. Valadares, Á. A. d. C. C. Sobrinho, and A. Perkusich, "Formal Verification of a Trusted Execution Environment-Based Architecture for IoT Applications," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 17199–17210, 2021.

[125] S. Pinto, T. Gomes, J. Pereira, J. Cabral, and A. Tavares, "IIoTEED: An Enhanced, Trusted Execution Environment for Industrial IoT Edge Devices," *IEEE Internet Computing*, vol. 21, no. 1, pp. 40–47, 2017.

[126] M. Akgün, E. U. Soykan, and G. Soykan, "A Privacy-Preserving Scheme for Smart Grid Using Trusted Execution Environment," *IEEE Access*, vol. 11, pp. 9182–9196, 2023.

[127] Y. Fan, S. Liu, G. Tan, X. Lin, G. Zhao, and J. Bai, "One Secure Access Scheme Based on Trusted Execution Environment," in *17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering*, New York, NY, USA, pp. 135–143, 2018.

[128] E. Feng, D. Feng, D. Du, Y. Xia, and H. Chen, "sNPU: Trusted Execution Environments on Integrated NPUs," in *51st Annual International Symposium on Computer Architecture (ISCA)*, Buenos Aires, Argentina, pp. 248–261, 2024.

[129] H. Xie, J. Zheng, T. He, S. We, and C. Hu, "TEBDS: A Trusted Execution Environment-and-Blockchain-supported IoT data sharing system," *Future Generation Computer Systems*, vol. 140, pp. 321–330, 2023.

[130] W. Jiang, E. Li, W. Zhou, Y. Yang, and T. Luo, "IoT Access Control Model Based on Blockchain and Trusted Execution Environment," *Processes*, vol. 11, no. 3, pp. 1–13, 2023.

[131] L. Kang, Y. Xue, W. Jia, X. Wang, J. Kim, C. Youn, M. J. Kang, H. J. Lim, B. Jacob, and J. Huang, "Iceclave: A Trusted Execution Environment for In-Storage Computing," in *54th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-54)*, pp. 199–211, 2021.

[132] F. Mo, A. S. Shamsabadi, K. Katevas, S. Demetriou, I. Leontiadis, A. Cavallaro, and H. Haddadi, "DarkneTZ: Towards Model Privacy at the Edge Using Trusted Execution Environments," in *18th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 161–174, 2020.

[133] N. C. Will, "A Privacy-Preserving Data Aggregation Scheme for Fog/Cloud-Enhanced IoT Applications Using a Trusted Execution Environment," in *2022 IEEE International Systems Conference (SysCon)*, pp. 1–5, 2022.

[134] A. M. Naseri, W. Lucia, M. Mannan, and A. Youssef, "On securing cloud-hosted cyber-physical systems using trusted execution environments," in *Proceedings of the 2021 IEEE International Conference on Autonomous Systems (ICAS)*, Aug. 2021, pp. 1–5, IEEE.

[135] S. Wang, D. Huo, H. Zhang, Y. Wang, C. Li, and F. Shao, "SSIDB: Secure sharing of IoT data on blockchain with CP-ABE and trusted environment assistance," in *Proceedings of the 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, Oct. 2024, pp. 2022–2029, IEEE.

[136] S. Ott, B. Orthen, A. Weidinger, J. Horsch, V. Nayani, and J. E. Ekberg, "MultiTEE: Distributing trusted execution environments," in *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, Jul. 2024, pp. 1617–1629.

[137] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[138] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[139] Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, "Federated learning meets multi-objective optimization," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2039–2051, 2022.

[140] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A Stackelberg game perspective," *IEEE Network Letters*, vol. 2, pp. 23–27, 2020.

[141] M. R. Sprague, A. Jalalirad, M. Scavuzzo, C. Capota, M. Neun, L. Do, and M. Kopp, "Asynchronous federated learning for geospatial applications," in *Proceedings of the ECML PKDD 2018 Workshops*, Dublin, Ireland, Sep. 2018, pp. 21–28. Cham, Switzerland: Springer International Publishing, 2019.

[142] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-IID data," in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, Dec. 2020, pp. 15–24.

[143] G. Bao and P. Guo, "Federated learning in cloud-edge collaborative architecture: key technologies, applications and challenges," *Journal of Cloud Computing*, vol. 11, no. 1, p. 94, 2022.

[144] V. P. Chellapandi, L. Yuan, C. G. Brinton, S. H. Żak, and Z. Wang, "Federated learning for connected and automated vehicles: A survey of existing approaches and challenges," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 119–137, 2023.

[145] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, June 2020.

[146] S. Mohammadi, M. Mohammadi, S. Sinaei, A. Balador, E. Nowroozi, F. Flammini, and M. Conti, "Balancing privacy and accuracy in federated learning for speech emotion recognition," *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 191–199, 2023.

[147] S. Mohammadi, A. Balador, S. Sinaei, and F. Flammini, "Balancing privacy and performance in federated learning: A systematic literature review on methods and metrics," *Journal of Parallel and Distributed Computing*, vol. 192, p. 104918, 2024.

[148] K. Chen and Q. Yang, *Privacy-Preserving Computing: For Big Data Analytics and AI*. Cambridge, U.K.: Cambridge University Press, 2023.

[149] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Applied Sciences*, vol. 8, no. 12, p. 2663, 2018.

[150] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–35, 2022.

[151] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, M. Hammoudeh, H. Karimipour, and G. Srivastava, "Block hunter: Federated learning for cyber threat hunting in blockchain-based iiot networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 8356–8366, 2022.

[152] A. Yazdinejad, A. Dehghantanha, G. Srivastava, H. Karimipour, and R. M. Parizi, "Hybrid privacy preserving federated learning against irregular users in next-generation Internet of Things," *Journal of Systems Architecture*, vol. 148, p. 103088, 2024.

[153] M. Song, Z. Wang, Z. Zhang, Y. Song, Q. Wang, J. Ren, and H. Qi, "Analyzing user-level privacy attack against federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2430–2444, 2020.

[154] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "FED-IIoT: A robust federated malware detection architecture in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8442–8452, 2020.

[155] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

[156] R. Shrestha, M. Mohammadi, S. Sinaei, A. Salcines, D. Pampliega, R. Clemente, A. L. Sanz, E. Nowroozi, and A. Lindgren, "Anomaly detection based on LSTM and autoencoders using federated learning in smart electric grid," *Journal of Parallel and Distributed Computing*, vol. 193, p. 104951, 2024.

[157] J. A. Alzubi, O. A. Alzubi, A. Singh, and M. Ramachandran, "Cloud-IIoT-based electronic health record privacy-preserving by CNN and blockchain-enabled federated learning," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1080–1087, 2022.

[158] Y. Xu, M. Z. A. Bhuiyan, T. Wang, X. Zhou, and A. K. Singh, "C-fdrl: Context-aware privacy-preserving offloading through federated deep reinforcement learning in cloud-enabled IoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1155–1164, 2022.

[159] J. So, C. He, C.-S. Yang, S. Li, Q. Yu, R. E. Ali, B. Guler, and S. Avestimehr, "LightSecAgg: a Lightweight and Versatile Design for Secure Aggregation in Federated Learning," in *Proceedings of Machine Learning and Systems*, 2022.

[160] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning (PMLR)*, pp. 5650–5659, 2018.

[161] X. Zhang, A. Fu, H. Wang, C. Zhou, and Z. Chen, "A privacy-preserving and verifiable federated learning scheme," in *IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020.

[162] L. Zhang, J. Xu, P. Vijayakumar, P. K. Sharma, and U. Ghosh, "Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 5, pp. 2864–2880, 2023.

[163] J. Qi, F. Lin, Z. Chen, C. Tang, R. Jia, and M. Li, "High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18378–18391, 2022.

[164] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10782–10793, 2020.

[165] O. Stan, V. Thouvenot, A. Boudguiga, K. Kapusta, M. Zuber, and R. Sirdey, "A secure federated learning: Analysis of different cryptographic tools," in *Proceedings of SECRYPT 2022—19th International Conference on Security and Cryptography*, vol. 1, Jul. 2022, pp. 669–674.

[166] T. Dai, L. Duan, Y. Jiang, Y. Li, F. Mei, and Y. Sun, "FORCE: Highly efficient four-party privacy-preserving machine learning on GPU," in *Proceedings of the Nordic Conference on Secure IT Systems*, Nov. 2023, pp. 330–349. Cham, Switzerland: Springer Nature.

[167] C. Zheng, L. Wang, Z. Xu, and H. Li, "Optimizing privacy in federated learning with MPC and differential privacy," in *Proceedings of the 2024 3rd Asia Conference on Algorithms, Computing and Machine Learning*, Mar. 2024, pp. 165–169.

[168] Y. Xu, C. Peng, W. Tan, Y. Tian, M. Ma and K. Niu, "Non-interactive verifiable privacy-preserving federated learning," *Future Generation Computer Systems*, vol. 128, pp. 365–380, 2022.

[169] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2021.

[170] F. Colosimo and F. De Rango, "Median-krum: A joint distance-statistical based Byzantine-robust algorithm in federated learning," in *Proceedings of the Int'l ACM Symposium on Mobility Management and Wireless Access*, pp. 61–68, 2023.

[171] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[172] H. Fazli Khojir, D. Alhadidi, S. Rouhani, and N. Mohammed, "FedShare: secure aggregation based on additive secret sharing in federated learning," in *Proceedings of the 27th International Database Engineered Applications Symposium*, pp. 25–33, 2023.

[173] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[174] E. Nowroozi, I. Haider, R. Taheri, and M. Conti, "Federated Learning Under Attack: Exposing Vulnerabilities through Data Poisoning Attacks in Computer Networks," *IEEE Transactions on Network and Service Management*, 2024.

[175] E. Nowroozi, Y. Mekdad, M. H. Berenjestanaki, M. Conti, and A. El Fergougui, "Demystifying the transferability of adversarial attacks in computer networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3387–3400, 2022.

[176] K. Özfatura, E. Özfatura, A. Küpçü, and D. Gündüz, "Byzantines can also learn from history: Fall of centered clipping in federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 19, 2023.

[177] B. Tondi, A. Costanzo, and M. Barni, "Robust and large-payload DNN watermarking via fixed, distribution-optimized, weights," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[178] E. Nowroozi, M. Mohammadi, P. Golmohammadi, Y. Mekdad, M. Conti, and A. S. Uluagac, "Resisting deep learning models against adversarial attack transferability via feature randomization," *IEEE Transactions on Services Computing*, 2023.

[179] E. Nowroozi, M. Mohammadi, E. Savaş, Y. Mekdad, and M. Conti, "Employing deep ensemble learning for improving the security of

computer networks against adversarial attacks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 2096–2105, 2023.
[180] A. Triastcyn and B. Faltings, "Bayesian differential privacy for machine learning," *International Conference on Machine Learning (PMLR)*, pp. 9583–9592, 2020.
[181] X. Liu, H. Huang, and J. Xiang, "A personalized diagnosis method to detect faults in gears using numerical simulation and extreme learning machine," *Knowledge-Based Systems*, vol. 195, p. 105653, May 2020.
[182] H. Yu, K. Yuan, W. Li, N. Zhao, W. Chen, C. Huang, H. Chen, and M. Wang, "Improved Butterfly Optimizer-Configured Extreme Learning Machine for Fault Diagnosis," *Complexity*, vol. 2021, no. 1, p. 6315010, 2021.
[183] L. Zhang, G. Su, J. Yin, Y. Li, Q. Lin, X. Zhang, and L. Shao, "Bioinspired scene classification by deep active learning with remote sensing applications," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 5682–5694, 2021.
[184] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3–16, 2016.
[185] L. Zhao and J. Yu, "Evaluating DAG-based blockchains for IoT," in *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 507–513, IEEE, 2019.
[186] E. Nowroozi, R. Taheri, M. Hajizadeh, and T. Bauschert, "Verifying the robustness of machine learning based intrusion detection against adversarial perturbation," *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 9–15, 2024.
[187] E. Nowroozi, M. Mohammadi, and M. Conti, "An adversarial attack analysis on malicious advertisement URL detection framework," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1332–1344, 2022.
[188] E. Chu, I. Bang, S. H. Kim, and D. K. Sung, "Self-organizing and self-healing mechanisms in cooperative small-cell networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1576–1581, IEEE, 2013.
[189] E. M. Kandoussi, A. Houmairi, I. El Mir, and M. Bellafkih, "Enhancing cloud security: harnessing Bayesian game theory for a dynamic defense mechanism," *Cluster Computing*, pp. 1–18, 2024.
[190] L. Zhou, A. Diro, A. Saini, S. Kaisar, and P. C. Hiep, "Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges, and opportunities," *Journal of Information Security and Applications*, vol. 80, p. 103678, 2024.

Ahmad Rahdari received the B.Sc. degree in 2018 and the M.Sc. degree in 2022, both in Electrical Engineering from K. N. Toosi University of Technology (KNTU), Tehran, Iran. He was a researcher at Iran Telecommunication Research Center (ITRC) for one year, starting in 2022. For the time being, he is a Ph.D. student in Electrical Engineering - Communication Systems at Shiraz University, Shiraz, Iran. His current research interests include AI for security, adversarial machine learning, distributed systems, cryptography, information-theoretic security, and network security, particularly smart grids and IoT networks. He also serves as a reviewer for prominent journals, such as IEEE Transactions on Network and Service Management (IEEE TNSM), Computer Networks, Vehicular Communications, and Multimedia Tools and Applications.

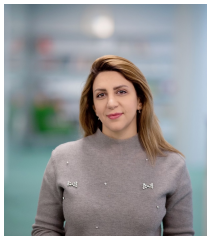
Elham Keshavarz holds a master's degree in Electrical Engineering with a specialization in Communication Systems. Since 2016, she has been a lecturer at Pishtazan Institute of Higher Education and Zand Institute of Higher Education in Shiraz, Iran, where she teaches a range of courses in Electrical and Computer Engineering. Her research interests focus on watermarking, machine learning, computer networks, image processing, and wireless and communication systems.

Ehsan Nowroozi, a Senior Member of IEEE, specializes in artificial intelligence for Cyber Security and is a Senior Lecturer at the Centre for Sustainable Cyber Security, University of Greenwich, London, United Kingdom. He completed his Ph.D. at the University of Siena, Italy, and held postdoctoral positions at the Universities of Siena and Padua in Italy, Sabanci University in Turkey, and Queen's University Belfast in the UK. He was also a Senior Lecturer at Ravensbourne University, London. His research focuses on multimedia forensics and security, AI verification, digital forensics, and machine learning security. He serves also as an associate editor at the IEEE Transactions on Network and Service Management (IEEE TNSM) since 2024, and technical member of IEEE Information Security (IEEE IFS).

Rahim Taheri received his Ph.D. degree in Information Technology from Shiraz University of Technology, Iran, in 2020. Now he is a Senior Lecturer in Cyber Security and Forensics at the University of Portsmouth, UK. Before joining the University of Portsmouth, he was a postdoctoral research associate at King's Communications, Learning, and Information Processing (Kclip) Lab, King's College London, UK. His main research interests include machine learning applications in security, adversarial machine learning, and federated learning.

Mehrdad Hajizadeh holds a Master's degree in Information and Communication Systems from the Technical University of Chemnitz, Germany. He is currently pursuing a Ph.D. with a research focus on offensive cybersecurity and adversarial machine learning. His work adopts cutting-edge generative AI technologies, such as GANs and large language models, to solve complex cybersecurity problems. Passionate about defending against cyber threats, he specializes in malware traffic analysis and network-based intrusion detection systems.

Mohammadreza Mohammadi holds a Master's degree in ICT and works as an AI R&D Engineer at RISE Research Institute of Sweden AB. His research focuses on federated learning, deep learning security and privacy, and applied AI. With a strong background in cutting-edge technologies in the field of AI, he is dedicated to advancing the field through innovative research and practical applications. His work aims to bridge the gap between theoretical research and real-world implementations in AI and machine learning.



Sima Sinaei (Member, IEEE) works as a Senior AI researcher at RISE Research Institutes of Sweden. Her research interests encompass Machine Learning, Deep Learning, Neural Network Architecture Optimization, Distributed AI Systems, and Federated Learning. Her primary focus is on EdgeAI, where she merges advancements in machine learning algorithms and systems with the development of optimized embedded computing platforms. This fusion aims to enable future AI applications at the edge, spanning across various industrial domains such as transportation, digital life, autonomous vehicles, and wearable healthcare applications.

Thomas Bauschert received the Dipl.-Ing. and Dr.-Ing. degrees from Technical University of Munich (TUM) in 1990 and 1997, respectively. From 1997 to 2007 he was with Siemens and Nokia Siemens Networks in Munich and since 2007 he is a Full Professor at the Technical University of Chemnitz (TUC), heading the Chair of Communication Networks at the Faculty of Electrical Engineering and Information Technology. His research interests are methods and architectures for flexible and reliable communication in fixed and wireless networks with a special focus on network design and automation as well as on network security.