Secure Multi-Party Computation for Efficient Deep Learning Inference

Arsenio Procacci
Department of Computer Science, George Mason University, Fairfax, VA, USA.
arsenio.procacci@gmu.edu

Kaj Lundstrom
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA.
kaj925@unr.edu

Himawan Astuti
Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL, USA.
hastuti@uab.edu

Colm Conmy
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
colm58@oregonstate.edu

Abstract

Secure multi-party computation offers a cryptographic foundation for deep learning inference in settings where inputs, models, or both must remain confidential. This paper examines the system-level design of efficient private inference from an interdisciplinary perspective that joins cryptographic protocol research, machine learning systems engineering, and socio-technical governance. The analysis begins with threat models and trust assumptions, then reviews major architectural families, including secret sharing, garbled circuits, homomorphic encryption, and hybrid protocols. It argues that performance limits are not solely algorithmic; they arise from memory movement, communication topology, nonlinear activation handling, deployment orchestration, and economic constraints. The paper analyzes how activation replacement, operator fusion, quantization-aware private computation, and hardware acceleration reshape structural trade-offs among latency, accuracy, privacy, robustness, and auditability. It also considers fairness and sustainability implications when private inference is deployed in healthcare, finance, public services, and edge environments. A central claim is that secure multi-party computation should be treated as an infrastructural capability rather than a drop-in cryptographic layer. Efficient deployment requires standard interfaces, reproducible benchmarks, governance for key management and collusion resistance, and lifecycle accounting for energy and maintenance. Case comparisons show that cross-domain differences in regulatory pressure, data sensitivity, and latency tolerance produce divergent optimal architectures. The paper concludes with a research agenda for composable protocols, adaptive privacy budgets, resilient service topologies, and policy frameworks that align cryptographic guarantees with institutional accountability.

Keywords

secure multi-party computation; private inference; deep learning systems; privacy-preserving machine learning; cryptographic acceleration; distributed trust; responsible deployment; socio-technical governance.

1. Introduction

Deep learning inference has become an infrastructural component of contemporary digital services, from medical triage and financial risk assessment to public-sector benefits administration and industrial monitoring. As these systems move into domains with sensitive data, the confidentiality of user inputs, proprietary model weights, and intermediate representations becomes a first-order design concern. Secure multi-party computation presents a rigorous approach to private inference by allowing multiple parties to jointly evaluate a model without revealing their respective inputs to one another. Early systems demonstrated that neural network prediction could be performed under cryptographic protection, but they also exposed steep performance penalties and complex trust assumptions [1]. Subsequent work has sought to reduce those penalties through hybrid protocol design, improved operator support, and systems-level co-design [2]. The resulting research field is no longer merely a cryptographic exercise; it is an infrastructure problem involving distributed trust, hardware acceleration, service orchestration, regulatory compliance, and environmental cost.

This paper adopts a system-level orientation. It does not treat secure multi-party computation as a single algorithm that can be inserted into an existing machine learning pipeline. Instead, it examines how protocol choices interact with model architecture, deployment topology, data governance, and institutional accountability. Private inference systems must satisfy multiple objectives simultaneously: they must protect confidentiality against defined adversaries, preserve predictive utility, achieve acceptable latency and throughput, remain robust under failure or collusion, and produce enough evidence for audit and oversight. These objectives often conflict. Stronger adversary models require more communication or computation, while aggressive optimization can weaken privacy guarantees or introduce new failure modes. The design space therefore resembles a socio-technical architecture problem rather than a narrow optimization problem.

The analysis proceeds through several connected themes. It first reviews foundations and threat models, then compares secure multi-party computation architectures for deep learning inference. It then examines efficiency bottlenecks and structural trade-offs, with attention to nonlinear operations, memory movement, and hardware constraints. Later sections address robustness, fairness, deployment governance, sustainability, and policy implications. Case illustrations from healthcare, finance, public services, and edge computing show that there is no universally optimal private inference architecture. Instead, the appropriate design depends on the sensitivity of data, the distribution of trust, the tolerance for latency, the regulatory environment, and the long-term operational capacity of the deploying institution. The paper concludes with a research agenda that emphasizes composability, benchmarking, lifecycle governance, and responsible deployment.

2. Foundations and Threat Models for Private Deep Learning Inference

Secure multi-party computation is grounded in the idea that mutually distrusting parties can compute a joint function while revealing only the output. Foundational results established that general secure computation is possible under cryptographic assumptions [11], and subsequent work formalized protocol composition and security in complex environments [10]. In the

context of deep learning inference, the joint function is usually a trained model that produces a prediction. The parties may include a model owner, a data owner, and one or more inference servers. Their interests are not identical. The model owner may wish to protect proprietary weights and architecture details, while the data owner may wish to protect personal or regulated data. The inference server may be a commercial cloud provider, a consortium node, or an edge device with limited resources. Each participant may have different incentives to deviate, collude, or infer information from protocol transcripts.

Threat models in private inference range from honest-but-curious adversaries to malicious participants who may arbitrarily deviate from the protocol. Honest-but-curious security assumes that parties follow the protocol but attempt to learn additional information from messages. Malicious security additionally requires correctness and privacy even when parties tamper with messages, abort strategically, or collude. Achieving malicious security is substantially more expensive in computation and communication, and it often requires additional mechanisms for consistency checks, commitment, and abort handling. The choice of threat model is therefore not merely a cryptographic detail; it determines the feasible deployment environment. A hospital consortium may tolerate a semi-honest model if all nodes are governed by a common legal framework, whereas an open marketplace may require stronger guarantees because participants are competitors.

Trust topology is another foundational dimension. Some protocols assume two non-colluding servers, while others rely on three or more parties with a threshold of honest participants. Secret sharing distributes a secret across parties so that a threshold can reconstruct it while fewer than the threshold learn nothing [13]. Beaver triples and related preprocessing techniques enable efficient multiplication within secret-shared representations [14]. Homomorphic encryption allows computation on encrypted data but often imposes high computational overhead and limited support for nonlinear operations [15]. Garbled circuits provide a different route to secure two-party computation by representing a function as an encrypted circuit [12]. Hybrid systems combine these approaches so that linear operations use one protocol family and nonlinear operations use another [2]. The practical consequence is that trust assumptions, communication patterns, and performance profiles are tightly coupled.

Deep learning inference adds domain-specific complications. Neural networks contain both linear operations, such as convolutions and matrix multiplications, and nonlinear activations, such as rectified linear units and sigmoid functions. Linear operations are relatively amenable to secure computation because they can be expressed as additions and multiplications. Nonlinear operations are more difficult because they require comparisons, branching, or approximations that are expensive under encryption or secret sharing. Differential privacy can complement secure computation by limiting information leakage through outputs, but it does not replace cryptographic protection of inputs and model weights [17]. Federated learning similarly addresses decentralized training but does not by itself guarantee secure inference, because gradients or model updates may still leak information [16]. A robust private inference architecture must therefore specify what is protected, from whom, under what assumptions, and for how long.

3. Secure Multi-Party Computation Architectures for Deep Learning Inference

Architectural choices in private inference can be understood through the balance between communication rounds, computational cost, and trust distribution. Two-party garbled circuit approaches offer strong security under appropriate assumptions but can be communication-intensive for large networks. Three-party secret-sharing protocols often achieve better

performance for neural network inference because they can exploit efficient preprocessing and parallel communication. Systems such as SecureNN demonstrated that three-party computation can support neural network training and inference with practical performance for certain workloads [8]. Later actively secure primitives over rings extended these ideas to stronger adversary models and private machine learning applications [9]. Frameworks such as CrypTen sought to bridge cryptographic protocols and machine learning abstractions, making secure computation more accessible to practitioners [7]. These developments show that private inference is gradually becoming a systems discipline with reusable layers rather than a collection of bespoke cryptographic demonstrations.

Hybrid architectures are particularly important for deep learning because different operators have different secure computation costs. SecureML introduced scalable privacy-preserving machine learning using hybrid protocols for linear and nonlinear operations [1]. Chameleon provided a hybrid framework that adapts secure computation techniques to machine learning applications [2]. GAZELLE focused on low-latency secure neural network inference by combining homomorphic encryption and garbled circuits [3]. MiniONN explored oblivious neural network predictions through transformations that reduce the cost of nonlinear layers [4]. Delphi proposed a cryptographic inference service that balanced latency and accuracy through protocol and architecture co-design [5]. Together, these systems illustrate that no single cryptographic primitive dominates. The optimal architecture depends on the network topology, the number of parties, the adversary model, and the deployment environment.

A recurring challenge is the treatment of nonlinear activation functions. Private inference systems often replace or approximate nonlinearities to make them compatible with efficient secure computation. For instance, activation replacement strategies have been developed to reduce the cost of nonlinear operations during private inference [6]. Such transformations can significantly improve latency, but they may also alter model behavior, require retraining, or introduce accuracy loss. The trade-off is not only technical. In regulated domains, replacing an activation function may affect model interpretability, validation evidence, and post-deployment monitoring. A healthcare institution may need to demonstrate that an approximate private inference pipeline preserves clinical safety, while a financial institution may need to show that the altered model still satisfies fair-lending requirements. Therefore, architecture choices interact with governance and accountability.

Protocol composition is another architectural concern. Secure computation protocols are typically designed and analyzed in isolation, but real inference services compose multiple protocols across layers, batches, and client sessions. Composition theorems provide a foundation for reasoning about security when protocols are combined [10], but practical systems must also manage state, session keys, timeouts, and failure recovery. Machine learning frameworks such as TensorFlow and PyTorch provide abstractions for model execution, but they were not originally designed for cryptographic privacy guarantees [21][22]. Integrating secure computation into these frameworks requires careful treatment of data types, operator dispatch, memory layout, and distributed execution. The result is a layered architecture in which cryptographic primitives, machine learning compilers, runtime schedulers, and deployment controllers must be co-designed.

4. Efficiency Bottlenecks and Structural Trade-offs

Efficiency in private inference is often framed as a latency problem, but it is better understood as a multi-dimensional systems problem. Latency matters for interactive applications, throughput matters for batch services, and cost matters for sustainable operation.

Communication rounds are frequently the dominant bottleneck because wide-area networks introduce delays that cannot be eliminated by faster computation. Memory movement is another bottleneck: secret-shared values must be converted, reshaped, and synchronized across parties, which can consume more time than the cryptographic operations themselves. Hardware acceleration can improve local computation, but it does not automatically reduce network overhead. Consequently, the most effective optimizations often target communication topology, operator fusion, and batching rather than isolated arithmetic.

Nonlinear operations are a persistent source of inefficiency. Comparisons, exponentials, and branching are expensive in secret-shared or encrypted domains. Activation replacement can reduce this cost, as noted in recent work on automatic rectified linear unit replacement [6]. However, replacement strategies must be evaluated against accuracy, robustness, and fairness. A model that is optimized for private inference may behave differently on underrepresented groups or rare cases. If the replacement is performed automatically, the interaction between optimization objectives and fairness constraints can become opaque. This is not an argument against optimization; it is an argument for treating private inference as a lifecycle process that includes validation, monitoring, and redress. The same systems that improve efficiency can also create hidden technical debt if their assumptions are not documented and maintained [20].

Quantization and pruning are common techniques for reducing model size and computation. In private inference, they can also reduce communication because smaller representations require fewer bits to transmit. Yet quantization interacts with security. Some protocols rely on fixed-point or integer arithmetic, while others support floating-point approximations. The choice of numerical representation affects both cryptographic correctness and model accuracy. A system that quantizes aggressively may achieve lower latency but produce predictions that are harder to explain or defend. Conversely, a system that preserves high precision may be too slow for real-time use. The structural trade-off is therefore not simply accuracy versus speed; it is a negotiation among precision, privacy, robustness, and institutional trust.

Trust assumptions also shape efficiency. Protocols that assume non-colluding servers can be faster because they avoid expensive malicious-security checks. Protocols that tolerate collusion require additional redundancy and verification, which increases cost. The number of parties affects communication patterns: two-party protocols may have simpler trust assumptions but higher communication complexity for certain operations, while three-party protocols can distribute work more efficiently. The geographical distribution of parties matters as well. If parties are in different jurisdictions, legal constraints may prevent certain data flows, even if the protocol is technically feasible. Efficiency is therefore not a purely technical metric; it is embedded in legal, organizational, and geopolitical structures.

5. System-Level Optimization, Robustness, and Fairness

System-level optimization for private inference must consider the entire pipeline, from data ingestion to model serving. Operator fusion can reduce the number of secure computation rounds by combining adjacent operations into a single protocol invocation. Batching can improve throughput by amortizing communication costs across multiple queries. Scheduling can prioritize latency-sensitive requests while deferring less urgent computations. These optimizations resemble traditional distributed systems engineering, but they are constrained by cryptographic security requirements. For example, batching must not allow one client to infer information about another client's input. Scheduling must not create timing side channels that reveal sensitive attributes. Caching must not retain secret-shared values beyond their

permitted lifetime. The systems designer must therefore reason about security properties at every layer, not only at the cryptographic core.

Robustness is closely related to system design. A private inference service must continue to function under node failures, network partitions, and denial-of-service attacks. If a protocol requires all parties to be online, a single failure can halt inference. Threshold secret sharing can tolerate some failures, but it introduces questions about key management, participant authentication, and recovery. Malicious participants may attempt to corrupt results or learn information by colluding with others. Active security mechanisms can detect or prevent such behavior, but they add overhead. In practice, robustness is often achieved through redundancy and monitoring, but these measures must be designed so that they do not undermine privacy. For example, logging every protocol message for debugging may create a new confidentiality risk. Governance must specify what is logged, who can access logs, and how long logs are retained.

Fairness and privacy are sometimes treated as separate concerns, but they intersect in private inference. Privacy protections may prevent the collection of demographic data needed to audit fairness, creating a tension between confidentiality and accountability [18]. Differential privacy can limit disclosure, but it may also reduce the precision of fairness assessments [17]. Federated learning can keep raw data local, but it may still produce models that disadvantage certain groups if the training data are unrepresentative [16]. A private inference system that hides model weights may also hide the evidence needed to detect bias. Ethical frameworks for algorithms emphasize transparency, accountability, and contestability, but these values must be translated into technical mechanisms [19]. Possible approaches include privacy-preserving audits, secure aggregation of fairness metrics, and multi-party governance boards that can authorize limited disclosures under controlled conditions.

The social implications of private inference are significant. On one hand, secure computation can enable beneficial data sharing in medicine and public health, where privacy concerns often block collaboration. On the other hand, it can also entrench opaque decision-making by shielding proprietary models from scrutiny. The same cryptographic guarantees that protect patients may protect institutions from accountability. This dual character means that private inference should not be framed as inherently ethical. It is a capability that can be used well or poorly. Its governance must address purpose limitation, data minimization, redress, and the distribution of power among parties. Technical design choices, such as who holds shares, who authorizes reconstruction, and who can audit protocol transcripts, are also political choices about trust and control.

6. Deployment, Governance, Sustainability, and Policy Implications

Deployment of private inference requires an operational model that extends beyond cryptographic protocols. Organizations must manage key material, authenticate participants, provision hardware, monitor performance, and respond to incidents. Key management is especially challenging because the compromise of a key or a threshold of shares can break confidentiality. Governance must specify how keys are generated, rotated, recovered, and destroyed. It must also address collusion risk: if parties are legally or economically entangled, the assumption of non-collusion may be unrealistic. In some cases, a neutral third party or a consortium governance structure may be necessary to maintain trust. These arrangements are not purely technical; they are institutional designs that allocate authority and responsibility.

Policy frameworks for private inference are still emerging. Data protection regulations often emphasize purpose limitation, storage limitation, and individual rights, but they may not directly address cryptographic computation across multiple parties. Sector-specific rules in healthcare and finance impose additional constraints on auditability, reporting, and model validation. Cross-border data flows can be restricted even when data remain encrypted, because regulators may treat any transfer of shares as a transfer of data. Policymakers must therefore develop guidance that distinguishes between raw data, encrypted shares, and derived outputs. At the same time, they must avoid creating loopholes that allow organizations to evade accountability by claiming cryptographic protection. A balanced approach would require documented threat models, independent audits, and clear lines of responsibility.

Sustainability is another underappreciated dimension. Private inference often requires more computation and communication than plaintext inference. This additional cost has energy and carbon implications, especially if protocols use many communication rounds across distributed data centers. The environmental burden may be justified when privacy is essential, but it should be measured and managed. System designers can reduce energy use through efficient scheduling, hardware acceleration, and protocol selection. They can also choose deployment topologies that minimize unnecessary data movement. Sustainability metrics should be included in benchmarks alongside latency and accuracy. Otherwise, private inference may become a technically impressive but environmentally costly solution that is difficult to scale responsibly.

Governance also involves human factors. Operators need training to understand the limits of cryptographic guarantees. Developers need tools that make secure computation abstractions usable without requiring deep expertise in cryptography. Auditors need access to standardized evidence about protocol configuration and security assumptions. Users need clear explanations of what privacy protection means in practice. These requirements point toward a socio-technical infrastructure in which technical standards, organizational processes, and legal rules co-evolve. The alternative is a fragmented landscape in which each deployment invents its own informal assurances. Such fragmentation increases risk, raises costs, and undermines public trust.

7. Case Illustrations and Cross-Domain Comparisons

Healthcare provides a compelling case for private inference. Hospitals may wish to collaborate on diagnostic models without sharing patient records. A secure multi-party computation system could allow a model owner to offer inference services while hospitals retain control over patient data. Latency requirements vary: emergency triage may demand rapid responses, while retrospective analysis can tolerate longer computation. Regulatory pressure is high, so auditability and validation are essential. A protocol that replaces nonlinear activations to improve efficiency must be carefully evaluated for clinical safety [6]. The trust model may involve a consortium of hospitals and a technology vendor, with governance rules for key management and incident response. In this setting, privacy is not optional, but it must be balanced against safety and equity.

Finance presents a different profile. Financial institutions handle sensitive personal and transactional data, and they operate under strict supervision. Private inference could enable fraud detection across institutions without revealing customer records. However, competitive dynamics may make parties reluctant to share even secret-shared information. Latency requirements are often moderate, but throughput can be high. Malicious security may be necessary because participants have strong incentives to game the system. The deployment

may rely on a small number of trusted nodes with hardware security modules, or on a consortium with independent governance. Fairness concerns are also acute because credit and fraud models are subject to anti-discrimination law. Private inference systems must therefore support auditing and explainability without compromising confidentiality [18].

Public services illustrate the tension between efficiency and accountability. Governments may use private inference to deliver benefits, detect fraud, or allocate resources. Citizens may benefit from privacy protections, but they may also demand transparency about automated decisions. A private inference system that hides model weights and decision logic may make it difficult to contest erroneous outcomes. Ethical frameworks emphasize the importance of explanation, appeal, and human oversight [19]. In public-sector deployments, governance must be especially robust, with independent oversight and clear legal authority. The technical architecture may include multiple parties, such as a government agency, a vendor, and a civil society auditor. The distribution of shares and the rules for reconstruction become matters of public concern.

Edge computing introduces resource constraints that shape private inference architecture. Mobile devices and sensors may have limited computation, memory, and battery life. Secure computation may be split between the edge and the cloud, with lightweight operations performed locally and heavier operations outsourced. This creates new trust questions: the edge device may be physically accessible to an adversary, while the cloud may be subject to subpoena or insider threats. Protocols must be efficient enough to avoid draining batteries, and they must tolerate intermittent connectivity. The choice of activation replacement, quantization, and batching must account for these constraints [6]. Edge deployments also raise sustainability issues because millions of devices can consume significant energy over time. Efficiency is therefore not only a performance goal but also an environmental and operational necessity.

8. Future Directions and Research Agenda

Future research on secure multi-party computation for deep learning inference should prioritize composability. Many current systems are designed as monolithic pipelines, but real deployments require modular components that can be replaced as cryptographic techniques and hardware evolve. Standards for operator interfaces, security parameters, and audit logs would reduce integration costs and improve trust. Research should also address adaptive privacy budgets. Different queries may require different levels of protection, and the system should be able to adjust protocol strength, batching, and disclosure controls accordingly. This requires new methods for reasoning about cumulative leakage across sessions and parties [17].

Another priority is benchmarking. The field needs reproducible benchmarks that measure latency, throughput, communication cost, energy use, accuracy, fairness, and security assumptions. Benchmarks should include realistic network conditions, failure scenarios, and adversarial participants. They should also cover diverse model architectures and deployment topologies. Without shared benchmarks, claims about efficiency are difficult to compare and easy to overstate. Standardized benchmarks would help researchers, practitioners, and regulators evaluate trade-offs more transparently. They would also support procurement decisions by providing comparable evidence about system performance and risk.

Hardware acceleration is likely to play an increasingly important role. Specialized processors for cryptographic operations, secure enclaves, and memory-safe accelerators can reduce the cost of private inference. However, hardware introduces supply-chain risks, vendor lock-in,

and new attack surfaces. Research should explore heterogeneous architectures that combine trusted and untrusted components without weakening the overall security model. It should also consider how hardware acceleration affects sustainability. Faster computation may reduce energy per inference, but it may also encourage more frequent use, leading to rebound effects. Lifecycle assessment should be part of the design process.

Finally, governance must become a first-class research topic. Technical papers often assume that institutions will implement protocols correctly and that participants will behave according to the threat model. In practice, governance failures, misaligned incentives, and regulatory ambiguity can undermine even well-designed systems. Interdisciplinary research should examine how to structure consortium agreements, audit mechanisms, liability rules, and dispute resolution for private inference. It should also study how different communities perceive privacy and fairness, because these perceptions affect adoption and trust. The goal is not only to make private inference efficient, but also to make it legitimate, accountable, and sustainable.

9. Conclusion

Secure multi-party computation offers a powerful foundation for private deep learning inference, but its value depends on system-level design and governance. Cryptographic protocols alone cannot resolve the trade-offs among latency, accuracy, robustness, fairness, and sustainability. Efficient private inference requires careful coordination across model architecture, protocol selection, hardware acceleration, deployment topology, and institutional policy. Hybrid architectures, activation replacement, operator fusion, and batching can improve performance, but they must be evaluated against security assumptions and social objectives. Trust models must be explicit, and governance must address key management, collusion resistance, auditing, and redress. As private inference moves from research prototypes to production infrastructure, the field must develop shared benchmarks, composable standards, and interdisciplinary methods for accountability. The central challenge is not merely to compute securely, but to build private inference systems that are efficient, equitable, and trustworthy over their full lifecycle.

References

1. Mohassel, P., & Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 19-38). IEEE. https://doi.org/10.1109/SP.2017.12

2. Riazi, M. S., Weinert, C., Tkachenko, O., Songhori, E. M., Schneider, T., & Koushanfar, F. (2018). Chameleon: A hybrid secure computation framework for machine learning applications. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security (pp. 707-721). Association for Computing Machinery. https://doi.org/10.1145/3196494.3196522

3. Juvekar, C., Vaikuntanathan, V., & Chandrakasan, A. (2018). GAZELLE: A low latency framework for secure neural network inference. In 27th USENIX Security Symposium (pp. 1651-1669). USENIX Association.

4. Liu, J., Juuti, M., Lu, Y., & Asokan, N. (2017). Oblivious neural network predictions via MiniONN transformations. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 619-631). Association for Computing Machinery. https://doi.org/10.1145/3133956.3134056

5. Mishra, P., Lehmkuhl, R., Srinivasan, A., Zheng, W., & Popa, R. A. (2020). Delphi: A cryptographic inference service for neural networks. In 29th USENIX Security Symposium (pp. 2505-2522). USENIX Association.

6. Peng, H., Huang, S., Zhou, T., Luo, Y., Wang, C., Wang, Z., ... & Ding, C. (2023, October). Autorep: Automatic relu replacement for fast private network inference. In 2023 IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 5155-5165). IEEE.

7. Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., & Ibrahim, M. (2021). CrypTen: Secure multi-party computation meets machine learning. In Advances in Neural Information Processing Systems 34 (pp. 4961-4973). Curran Associates.

8. Wagh, S., Gupta, D., & Chandran, N. (2019). SecureNN: 3-party secure computation for neural network training. Proceedings on Privacy Enhancing Technologies, 2019(3), 26-49. https://doi.org/10.2478/popets-2019-0035

9. Damgård, I., Escudero, D., Frederiksen, T., Keller, M., Scholl, P., & Volgushev, N. (2019). New primitives for actively-secure MPC over rings with applications to private machine learning. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 1102-1120). IEEE. https://doi.org/10.1109/SP.2019.00078

10. Canetti, R. (2000). Security and composition of multiparty cryptographic protocols. Journal of Cryptology, 13(1), 143-202. https://doi.org/10.1007/s001459910006

11. Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play any mental game. In Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (pp. 218-229). Association for Computing Machinery. https://doi.org/10.1145/28395.28420

12. Yao, A. C. (1982). Protocols for secure computations. In 23rd Annual Symposium on Foundations of Computer Science (pp. 160-164). IEEE. https://doi.org/10.1109/SFCS.1982.38

13. Shamir, A. (1979). How to share a secret. Communications of the ACM, 22(11), 612-613. https://doi.org/10.1145/359168.359176

14. Beaver, D. (1991). Efficient multiparty protocols using circuit randomization. In Advances in Cryptology - CRYPTO '91 (pp. 420-432). Springer. https://doi.org/10.1007/3-540-46766-1_34

15. Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In Advances in Cryptology - EUROCRYPT '99 (pp. 223-238). Springer. https://doi.org/10.1007/3-540-48910-X_16

16. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1-2), 1-210. https://doi.org/10.1561/2200000083

17. Dwork, C. (2006). Differential privacy. In Automata, Languages and Programming (pp. 1-12). Springer. https://doi.org/10.1007/11787006_1

18. Barocas, S., Hardt, M., & Narayanan, A. (2019). Fairness and machine learning: Limitations and opportunities. fairmlbook.org.

19. Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. Big Data & Society, 3(2), 1-21. https://doi.org/10.1177/2053951716679679

20. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden technical debt in machine learning systems. In Advances in Neural Information Processing Systems 28 (pp. 2503-2511). Curran Associates.

21. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (pp. 265-283). USENIX Association.

22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32 (pp. 8024-8035). Curran Associates.