

Communication-Efficient Two-Party Secure Computation for Deep Learning Inference

Armando Trentini

Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA.

armando.trentini993@ku.edu

Coriolano Asmundo

Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA.

coriolano.work@buffalo.edu

Eugenia Faust

Department of Computer Science, George Mason University, Fairfax, VA, USA.

eugeniafaust711@gmu.edu

Evangelos Romer

Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL, USA.

eromer@uab.edu

Abstract

Deep learning inference increasingly crosses organizational, jurisdictional, and device boundaries, creating demand for cryptographic methods that protect both client inputs and model parameters. Two-party secure computation offers a principled way to evaluate neural networks without revealing either party's private data, yet practical deployment remains constrained by communication cost, latency instability, infrastructure heterogeneity, and governance complexity. This paper examines communication-efficient two-party secure computation for deep learning inference from a system-level perspective rather than a narrow cryptographic one. It analyzes how protocol choice, network topology, preprocessing strategy, model architecture, and deployment environment interact to shape feasibility, security, and fairness. The discussion considers secure multiparty computation foundations, hybrid protocol composition, homomorphic encryption, differential privacy, and secure aggregation, with attention to their operational trade-offs. It then connects these technical choices to governance, auditability, policy compliance, sustainability, and economic viability. Particular emphasis is placed on the structural tension between reducing communication rounds and preserving strong privacy guarantees under realistic adversarial conditions. The paper also considers robustness, fairness, and redress in private inference systems, arguing that cryptographic protection alone cannot resolve broader questions of accountability and distributive impact. Through cross-domain comparisons in healthcare, finance, mobile computing, and public-sector services, the analysis identifies architectural patterns that may improve deployability without overstating security. The conclusion proposes an interdisciplinary research agenda that integrates protocol design, infrastructure planning, regulatory alignment, and human-centered evaluation for communication-efficient private inference.

Keywords

communication-efficient secure computation; two-party computation; private deep learning inference; secure multiparty computation; system architecture; privacy governance; deployment sustainability.

1. Introduction

Deep learning inference has become a foundational computational service, embedded in diagnostic support, financial risk scoring, recommendation, identity verification, and public-sector decision-making. In many of these settings, the entity that holds the model and the entity that holds the input are distinct, and neither can fully trust the other with raw data. A hospital may wish to query a specialized diagnostic model without disclosing patient records, while the model provider may wish to protect proprietary weights and training investments. A mobile user may wish to obtain a prediction without uploading sensitive keystrokes or images. These scenarios have motivated private inference systems that combine cryptographic protection with machine learning execution. Two-party secure computation is especially relevant because it maps naturally onto the client-server relationship that dominates deployed inference services.

However, the path from cryptographic feasibility to operational viability is narrow. Early secure inference systems demonstrated that neural networks could be evaluated under secure multiparty computation, oblivious transfer, garbled circuits, and secret sharing, but they also revealed that communication, not computation alone, often becomes the binding constraint [1], [2], [3], [4]. Neural network inference involves repeated linear transformations and nonlinear activation functions, and the latter are particularly expensive in secure computation because they require comparisons, oblivious selection, or interactive protocols. Recent architectural work has attempted to reduce the number of such operations in order to lower communication and latency [5]. Yet communication efficiency cannot be treated as a purely algorithmic problem. It is shaped by wide-area network conditions, edge device constraints, cloud tenancy, preprocessing availability, trust assumptions, and regulatory obligations.

This paper argues that communication-efficient two-party secure computation for deep learning inference should be understood as a socio-technical infrastructure problem. The central question is not only whether a protocol can evaluate a network with acceptable cryptographic security, but whether the resulting system can be deployed equitably, audited credibly, sustained economically, and governed responsibly. The analysis proceeds through several layers. It first reviews the threat landscape and cryptographic foundations that define what two-party inference can promise. It then examines communication bottlenecks and architectural trade-offs, including hybrid protocol composition and preprocessing strategies. Next, it considers infrastructure, deployment, and governance concerns, followed by robustness, fairness, and policy implications. It closes with sustainability, cross-domain comparisons, and a forward-looking research agenda.

2. Foundations and Threat Landscape of Private Deep Learning Inference

Two-party secure computation builds on foundational results in secure function evaluation and secure multiparty computation. The earliest formulations established that two parties could jointly compute a function while revealing only the output, using oblivious transfer and circuit-based techniques [6]. General multiparty computation extended this idea to broader settings and provided formal security definitions under composition [7]. Subsequent work on circuit randomization and preprocessing enabled more efficient online phases, in which expensive cryptographic material is generated before the private inputs are known [8]. These

foundations remain essential because they clarify the security guarantees that private inference can and cannot provide. A protocol may protect input confidentiality and model confidentiality under a defined adversary model, but it does not automatically protect against inference from the output, coercion, side channels, or operational misuse.

The threat landscape for deep learning inference is multidimensional. In a semi-honest model, parties follow the protocol but attempt to learn additional information from messages. In a malicious model, parties may deviate arbitrarily, which requires stronger verification and consistency mechanisms. Model extraction, membership inference, attribute inference, and gradient leakage illustrate that machine learning systems create privacy risks that extend beyond raw data disclosure [19]. Differential privacy offers a complementary guarantee by limiting the influence of any individual record on outputs, but it introduces a privacy-utility trade-off that must be tuned carefully [12], [13]. Homomorphic encryption permits computation on encrypted data and can reduce interaction in some settings, but it often imposes high computational overhead and constrains the types of operations that can be evaluated efficiently [14], [15]. Federated learning and secure aggregation provide alternatives for distributed training and aggregation, but they do not solve all inference-time trust problems [16], [20].

For two-party inference, the most common trust configuration involves a client with private inputs and a server with a private model. The client may be a hospital, a mobile application, or a financial institution, while the server may be a cloud provider or specialized model vendor. Security goals typically include input privacy, model privacy, and output privacy, although the last is often selectively relaxed because the client must learn the prediction. The choice of adversary model determines whether the system can tolerate misbehavior, abort, or provide verifiable correctness. Composition matters because real systems combine multiple cryptographic protocols, each with different assumptions and failure modes [9], [10], [11]. A secure protocol embedded in a poorly governed service can still produce unfair, unauditible, or unsafe outcomes. Thus, the threat model must be connected to institutional accountability, not only to cryptographic definitions.

3. Communication Bottlenecks in Two-Party Inference Architectures

Communication cost in private inference arises from several interacting sources. Garbled circuits require transmitting encrypted truth tables and keys, while secret-sharing protocols require rounds of message exchange to perform multiplication and nonlinear operations. Homomorphic encryption can shift much of the communication burden to computation, but ciphertext expansion and key management remain significant. In practice, the communication profile depends on the network architecture, batch size, layer types, and the placement of nonlinearities. Convolutional layers, attention mechanisms, normalization layers, and activation functions each stress secure computation differently. Systems such as SecureML, MiniONN, Chameleon, and Gazelle illustrate different points in the design space, balancing preprocessing, garbled circuits, secret sharing, and homomorphic encryption [1], [2], [3], [4]. Their differences show that there is no single optimal protocol for all models or deployment contexts.

Nonlinear activation functions are a persistent bottleneck because secure comparison and selection are more expensive than secure addition or multiplication. Architectural responses such as ReLU-reduced network design have attempted to restructure model topologies to decrease the number and cost of nonlinear operations that dominate secure evaluation [5]. This approach is promising because it aligns model design with cryptographic cost models,

but it also raises questions about accuracy, generalization, and compatibility with pretrained models. A communication-efficient system may need to co-design the neural architecture, quantization strategy, and protocol schedule. For example, reducing activation precision can lower communication, but it may also reduce robustness and fairness if it disproportionately affects underrepresented groups. Similarly, batching improves throughput but increases memory pressure and may worsen latency for interactive applications.

Communication rounds are often more damaging than total bytes in wide-area networks. High round-trip time, packet loss, and variable bandwidth can make a protocol with low total data but many rounds slower than a protocol with fewer rounds and larger messages. This observation shifts attention from raw communication volume to round complexity, pipelining, and parallelism. Preprocessing can move expensive operations offline, but it requires storage, coordination, and assumptions about future inputs. In cloud deployments, preprocessing may be performed by a trusted dealer, a separate cryptographic service, or the two parties themselves. Each choice changes the trust model and the operational burden. Edge deployment introduces additional constraints, including intermittent connectivity, limited battery, and hardware heterogeneity. A communication-efficient design must therefore specify not only the protocol but also the deployment topology, failure recovery, and resource scheduling policies.

4. Architectural Trade-offs and Protocol Composition

Hybrid protocol composition is central to practical private inference. Different operations within a neural network may be evaluated using different cryptographic primitives, and the system must convert between representations without leaking information. Mixed-protocol frameworks demonstrate that switching between arithmetic sharing, Boolean sharing, and garbled circuits can improve performance, but conversion costs and security assumptions must be managed carefully [9], [10]. The online phase may rely on precomputed correlated randomness, while the offline phase generates Beaver triples or other auxiliary data [8]. If preprocessing is outsourced, the system must ensure that the preprocessing service cannot compromise privacy or correctness. If preprocessing is distributed, the system must tolerate network failures and avoid bottlenecks. These architectural decisions determine whether the system can sustain low latency under load.

Another trade-off concerns where the model and input reside. In some designs, the client holds the input and the server holds the model. In others, the model is split across parties, or the client performs part of the computation locally. Split inference can reduce communication by keeping early layers on the client, but it may leak information through intermediate representations. Conversely, server-side computation can protect the client device but increases bandwidth and dependency on centralized infrastructure. Secure aggregation and federated approaches suggest that distributed trust can reduce single-point risks, yet they introduce coordination overhead and may not support arbitrary inference workloads [16], [20]. The optimal architecture depends on the sensitivity of the input, the value of the model, the regulatory environment, and the acceptable latency.

Security level and verification also shape communication. Semi-honest protocols are generally more efficient, but they are insufficient when parties may deviate. Malicious-security protocols add consistency checks, commitments, or zero-knowledge proofs, which increase communication and computation. Verifiable computation can assure the client that the server executed the correct model, but it may reveal model details or require trusted hardware. Trusted execution environments offer another route, but they introduce hardware

trust assumptions and side-channel risks. A mature system should make these trade-offs explicit in service-level agreements and audit reports. The question is not whether a protocol is absolutely secure, but whether its assumptions match the threat model and the consequences of failure. This is a governance question as much as a cryptographic one.

5. Infrastructure, Deployment, and Operational Governance

Deploying communication-efficient private inference requires infrastructure that supports cryptographic operations, key management, secure logging, and policy enforcement. Cloud environments offer elasticity and geographic distribution, but they also create multi-tenancy risks, cross-border data flows, and dependency on providers. Edge environments reduce data movement but constrain computation and complicate updates. On-premises deployments can satisfy data residency requirements but may lack specialized hardware and operational expertise. A realistic system may combine these models, using edge devices for preprocessing or feature extraction, cloud resources for heavy cryptographic operations, and private data centers for regulated data. The architecture must therefore support heterogeneous execution, secure orchestration, and graceful degradation when network conditions deteriorate.

Operational governance is essential because private inference systems process sensitive data and influence consequential decisions. Key management must cover key generation, rotation, revocation, and recovery. Audit logs should record protocol executions, model versions, access policies, and anomalies without undermining privacy. Compliance with data protection regimes requires data minimization, purpose limitation, and user rights, but cryptographic protocols can complicate deletion and explanation. If computations are distributed, determining where data was processed and who is accountable becomes difficult. Governance frameworks must define roles for model owners, infrastructure providers, auditors, and regulators. Design science approaches can help structure these socio-technical systems by treating institutions, incentives, and technical artifacts as interrelated design objects [18].

Homomorphic encryption and secure aggregation illustrate the governance tension between protection and control. Homomorphic encryption can enable computation on encrypted data, but it may limit inspection and debugging, which can hinder fairness audits and incident response [14], [15]. Secure aggregation can protect individual contributions in federated settings, but it may also obscure malicious updates unless additional mechanisms are used [16]. Differential privacy can provide formal guarantees, but the choice of privacy budget affects utility and may be opaque to affected populations [12], [13]. A communication-efficient system should therefore include transparency reports, independent evaluation, and mechanisms for contestability. Without these, efficiency gains may be achieved at the cost of accountability.

6. Robustness, Fairness, and Policy Implications

Robustness in private inference includes resistance to adversarial inputs, distribution shift, network failures, and protocol misbehavior. Adversarial examples can cause misclassification regardless of cryptographic protection, and private inference may make it harder to inspect inputs for anomalies. Secure computation can hide the very data that a defender might use to detect manipulation. This creates a tension between privacy and security monitoring. A system may need privacy-preserving anomaly detection, secure logging, or policy-based disclosure under exceptional circumstances. The design must specify who can authorize disclosure, under what legal basis, and with what oversight. Otherwise, emergency access mechanisms may become vectors for abuse.

Fairness is equally important. A private inference system can produce discriminatory outcomes even if it perfectly protects data. If training data underrepresent certain groups, the model may encode historical bias, and cryptographic privacy does not correct that bias [17]. Communication-efficient design may exacerbate fairness problems if it encourages aggressive quantization, pruning, or activation reduction that harms minority subgroups. Fairness audits typically require access to demographic data and model behavior, but privacy protections may limit such access. Techniques such as secure aggregation and differential privacy can support audits, but they require careful design to avoid masking disparate impact. Policy frameworks should require impact assessments, bias testing, and redress mechanisms before private inference systems are deployed in high-stakes domains.

Policy implications extend beyond data protection. Governments may need to clarify liability when a private inference service causes harm, especially when the model provider, infrastructure provider, and client are different entities. Certification regimes could establish minimum security, fairness, and transparency standards. But overly prescriptive rules may freeze technology and favor large incumbents that can afford compliance. Regulators could instead promote interoperability, open standards, and independent auditing. They could also support public-interest infrastructure for privacy-preserving computation in healthcare, education, and social services. The goal should be to align cryptographic efficiency with democratic accountability, not to treat privacy as a purely technical checkbox.

7. Sustainability, Economics, and Cross-Domain Comparisons

Communication efficiency has sustainability implications. Cryptographic protocols can consume significant energy through computation, data transmission, and cooling. Moving computation to edge devices may reduce transmission but can shorten battery life and increase electronic waste. Cloud data centers may be more energy efficient per operation, but they concentrate demand and depend on regional energy mixes. Batching and preprocessing can improve efficiency, but they may require always-on infrastructure that consumes power even when idle. A sustainable design should consider lifecycle emissions, hardware utilization, and rebound effects. If private inference enables new applications that increase overall computation, aggregate energy use may rise even if per-inference cost falls.

Economically, two-party private inference must compete with less private alternatives. Clients may not pay a premium unless regulation, reputation, or contractual obligations require it. Model providers may resist protocols that expose weights or reduce throughput. Infrastructure providers may prefer proprietary services that lock in customers. These incentives shape what gets deployed. Open-source implementations, standardized interfaces, and hardware acceleration can lower barriers. Yet sustainability also depends on business models. Subscription pricing, usage-based fees, and public procurement could support privacy-preserving services. Cross-domain comparisons reveal different priorities. Healthcare emphasizes regulatory compliance and auditability. Finance emphasizes fraud detection, latency, and liability. Mobile computing emphasizes battery, bandwidth, and user consent. Public-sector services emphasize equity, transparency, and due process. A communication-efficient protocol must be adapted to each domain's institutional logic.

Federated learning and secure aggregation offer instructive comparisons. They distribute trust and reduce raw data centralization, but they introduce coordination costs and may not protect against all inference attacks [16], [19]. Homomorphic encryption offers strong protection but may be too slow for large models without hardware support [15]. Two-party secure computation sits between these extremes, offering strong privacy for interactive inference but

requiring careful management of communication and trust. Hybrid systems that combine multiple techniques may be most practical, but they also increase complexity and attack surface. The challenge is to design modular architectures in which security properties are composable, auditable, and understandable to non-specialists.

8. Case Illustrations and Forward-Looking Perspectives

Consider a hospital that wants to use a third-party diagnostic model without disclosing patient images. A communication-efficient two-party protocol could allow the hospital to encrypt or secret-share the image, interact with the model provider, and receive a prediction. If the model contains many nonlinear operations, preprocessing and model redesign may be necessary to meet clinical latency requirements [5]. The hospital must also ensure that the prediction is explainable enough for clinicians and contestable by patients. Governance must specify who can access audit logs and how model updates are approved. In this case, efficiency is not only a technical metric; it affects whether the system can be used during time-sensitive care.

In finance, a bank may want to query a fraud detection model without revealing transaction details. Latency and availability are critical, and false positives can harm customers. A private inference system must be robust to adversarial manipulation and must support audit trails for regulators. Differential privacy or secure aggregation could protect customer data during model improvement, but they may complicate investigation of fraudulent patterns [12], [13], [16]. The bank must balance privacy with legal obligations to report suspicious activity. This illustrates that communication-efficient protocols must be embedded in a broader compliance architecture, not deployed as standalone tools.

For mobile applications, a user may want on-device inference with cloud assistance. Splitting the model can reduce communication, but intermediate features may leak information. Secure computation can protect the features, but battery and bandwidth constraints are severe. Model compression, early exit, and activation reduction can help, but they may reduce accuracy and fairness. The system must also handle intermittent connectivity and device heterogeneity. Future research should explore adaptive protocols that select cryptographic parameters based on network conditions, battery state, and sensitivity of the request. Such adaptivity must not silently weaken privacy. It should be governed by clear policies and user-facing controls.

A forward-looking research agenda should integrate several strands. First, model architectures and cryptographic protocols should be co-designed to reduce communication without sacrificing accuracy, robustness, or fairness. Second, hardware acceleration and specialized networking can lower latency, but their supply chains and energy impacts must be assessed. Third, formal verification and compositional security analysis can help manage the complexity of hybrid protocols [10], [11]. Fourth, governance frameworks should support auditing, redress, and public participation. Fifth, deployment studies should measure real-world performance across diverse populations and infrastructures. Design science can provide a methodological bridge between technical artifacts and institutional arrangements [18]. The field should move beyond isolated benchmarks toward sustained, interdisciplinary evaluation.

9. Conclusion

Communication-efficient two-party secure computation for deep learning inference is a promising but demanding socio-technical undertaking. Cryptographic foundations provide meaningful privacy guarantees, and systems research has shown that hybrid protocols, preprocessing, model redesign, and hardware acceleration can reduce communication and

latency. Yet efficiency gains are not sufficient for responsible deployment. Real systems must contend with adversarial behavior, network variability, regulatory constraints, economic incentives, environmental costs, and fairness risks. The most credible path forward is not a single optimal protocol but a modular architecture that makes trade-offs explicit, composable, and auditable. Such an architecture should align cryptographic assumptions with governance requirements, support independent evaluation, and remain adaptable across domains. Private inference will only achieve lasting legitimacy if it protects data while also sustaining accountability, equity, and public trust.

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. 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). ACM. <https://doi.org/10.1145/3133956.3134056>
3. 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 Asia Conference on Computer and Communications Security (pp. 707–721). ACM. <https://doi.org/10.1145/3196494.3196522>
4. 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.
5. Peng, H., Zhou, S., Luo, Y., Xu, N., Duan, S., Ran, R., ... & Ding, C. (2023). Rrnet: Towards relu-reduced neural network for two-party computation based private inference. arXiv preprint arXiv:2302.02292.
6. Yao, A. C. (1986). How to generate and exchange secrets. In 27th Annual Symposium on Foundations of Computer Science (pp. 162–167). IEEE. <https://doi.org/10.1109/SFCS.1986.25>
7. 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). ACM. <https://doi.org/10.1145/28395.28420>
8. 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
9. Demmler, D., Schneider, T., & Zohner, M. (2015). ABY: A framework for efficient mixed-protocol secure two-party computation. In Proceedings of the 2015 Network and Distributed System Security Symposium. Internet Society. <https://doi.org/10.14722/ndss.2015.23113>
10. Evans, D., Kolesnikov, V., & Rosulek, M. (2018). A pragmatic introduction to secure multi-party computation. Foundations and Trends in Privacy and Security, 2(2-3), 70–246. <https://doi.org/10.1561/33000000017>

11. Canetti, R. (2000). Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1), 143–202. <https://doi.org/10.1007/s001459910006>
12. Dwork, C. (2006). Differential privacy. In *Automata, Languages and Programming* (pp. 1–12). Springer. https://doi.org/10.1007/11787006_1
13. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 308–318). ACM. <https://doi.org/10.1145/2976749.2978318>
14. Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016). CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proceedings of the 33rd International Conference on Machine Learning* (pp. 201–210). PMLR.
15. Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., & Liu, Y. (2020). BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning. In *2020 USENIX Annual Technical Conference* (pp. 493–506). USENIX Association.
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/10.1561/22000000083>
17. Barocas, S., Hardt, M., & Narayanan, A. (2019). Fairness and machine learning: Limitations and opportunities. fairmlbook.org.
18. Wieringa, R. J. (2014). Design science methodology for information systems and software engineering. Springer. <https://doi.org/10.1007/978-3-662-43839-8>
19. Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32* (pp. 14774–14784). Curran Associates.
20. 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.