PQC meets ML or AI: Exploring the Synergy of Machine Learning and Post-quantum Cryptography

Saleh Darzi[1] and Attila A Yavuz[1]

[1]University of South Florida

March 07, 2024

IEEE Security and Privacy Magazine

PQC meets ML or AI: Exploring the Synergy of Machine Learning and Post-quantum Cryptography

Saleh Darzi, *University of South Florida, Tampa, USA, salehdarzi@usf.edu*

Attila A. Yavuz, *University of South Florida, Tampa, USA, attilaayavuz@usf.edu*

Abstract—Artificial Intelligence and Machine Learning are widely integrated into real-world applications, facing security and privacy risks. The emergence of quantum computers poses a substantial threat to ML's long-term security. Our study delves into the intersection of ML with security and privacy in the post-quantum era, where Post-Quantum Cryptography meets ML/AI.

Machine Learning (ML), a subset of Artificial Intelligence (AI), executes tasks and learns from datasets without explicit programming. Given the vast amounts of available data, ML automates time-consuming tasks, allowing machines to learn, understand, and respond. This has led to ML's integration into numerous real-world applications, spanning natural language processing (e.g., ChatGPT), healthcare systems, financial services, recommendation systems, and more. It's worth noting that companies may also leverage ML for cost-effective outsourcing of tasks to cloud-based infrastructure, giving rise to the paradigm known as ML-as-a-Service (MLaaS). ML addresses problems broadly classified into four categories: classification (e.g., email spam detection), clustering (e.g., e-commerce), prediction/regression (e.g., stock market prediction), and decision making (e.g., self-driving cars). Learning occurs in centralized, distributed, or collaborative manners, with Federated Learning (FL) falling under distributed learning [1].

Numerous concerns arise during the training and utilization of ML, including but not limited to data-related issues (such as data quality and confidentiality), efficiency challenges (scalability, heavy computational overhead), training considerations (fairness, robustness), ethical aspects (regulations, privacy), and security vulnerabilities (privacy, integrity, adversarial attacks). Privacy issues stand out as a significant threat among the various concerns associated with ML, its applications, and services. Precisely, privacy in ML encompasses concerns about the confidentiality of collected, pre-processed, and output data for data owners, model and parameter privacy for model owners, and services (MLaaS) privacy for the deploying organization. These phases are tightly integrated, and a privacy breach in one phase can lead to security issues in others. For example, threats like model extraction attacks aim to obtain the model or an equivalent version by querying it with different inputs. Reconstruction attacks use the model's output to gather sensitive information about input data or attributes, while membership inference attacks seek to determine if specific data was used to train a model [1].

Various security attacks on ML, driven by the goal of obtaining private information, accessing sensitive data, or gaining an advantage over the model, impede the widespread deployment of ML in real-world applications. To address these security and privacy concerns, the concept of Privacy-Preserving Machine Learning (PPML) has been introduced, and in distributed settings, such as FL, extends to PPFL [2]. Note that, ML is not always a foe threatening our privacy; it can also serve as a valuable ally in enhancing security. ML functions as a security countermeasure by detecting attacks through the analysis of network data flow, thereby thwarting potential security threats.

PPML solutions typically involve algorithmic (hardware or software), encryption-based, or distributed computation-based approaches, which are inherently of cryptographic nature and relying on computationally challenging problems. However, the emergence and advancements in quantum computers pose a threat to data protection, user privacy, and system security in the PQ era, as quantum algorithms can potentially break classical cryptographic solutions. Post-Quantum Cryptography (PQC) addresses this challenge and is broadly categorized into Lattice-based, Code-based, Multivariate, Isogenies, Symmetric Key-based, and

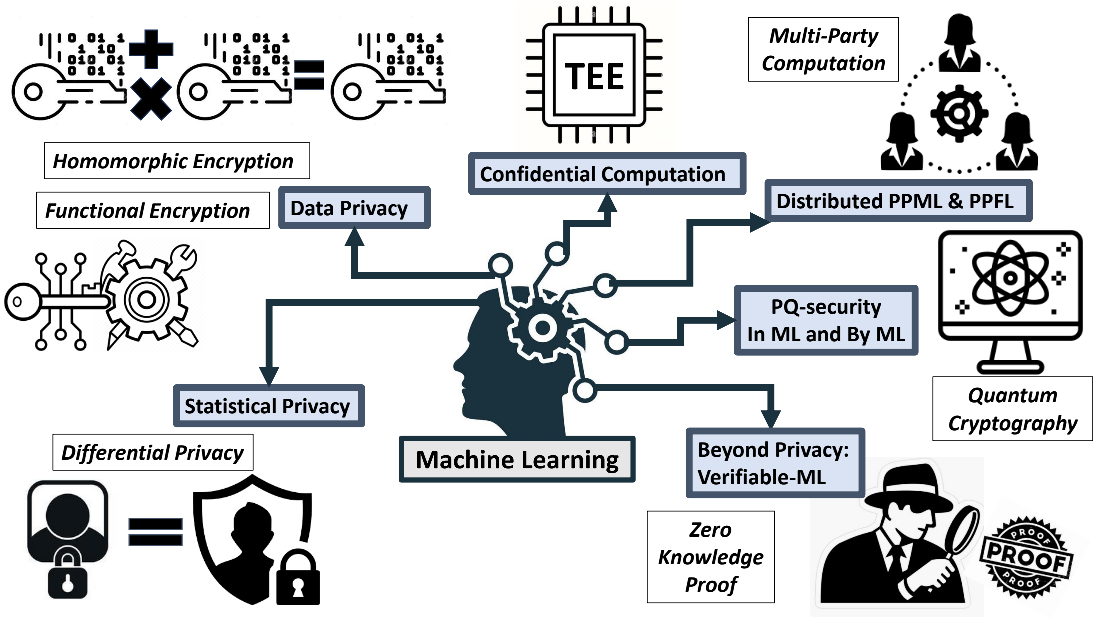

FIGURE 1. Various Aspects of ML/AI in Post-Quantum Era

Quantum Cryptography (QC), all grounded in mathematically hard problems while QC incorporating properties derived from quantum physics. Numerous initiatives, competitions, and projects in academia and industry are focused on realizing and implementing general-purpose PQC use-cases, with NIST taking a leading role in standardization efforts [3].

Securing Long-Term Trust: ML in PQ Era

In essence, these two pivotal domains, ML and PQC, have largely operated in isolation, with only a limited focus on integrating them to ensure long-term security and privacy. It's crucial to emphasize that, given the current extensive use of ML in numerous real-world applications, particularly those dealing with highly sensitive user information such as healthcare systems and financial services, there is a pressing need for a swift transition to Post-Quantum (PQ)-secure PPML solutions. Delaying this transition could lead to substantial long-term security threats in the near future. Thereby, several research gaps and potential solutions demand attention and resolution from academia and industry.

This article provides a thorough survey of potential security and privacy aspects of ML in the PQ era. It's crucial to emphasize that there is no one-size-fits-all methodology for achieving an ideal ML system that comprehensively addresses all aspects, including privacy solutions like PPML, security countermeasures for tasks such as network intrusion detection, integrity through verifiable ML, and perfect secrecy achieved by quantum, information theoretic, and hardware-based approaches. Different challenges and considerations exist for each facet of ML where Figure 1 illustrates the high-level aspects of all these PQ solutions. The detailed contribution of this article as follows:

- Note that the approaches in Figure 1 provide classical security levels and PQ promises. Here, we scrutinize their security, with a specific focus on their PQ properties and current state.
- We discuss provably secure methods, including Homomorphic Encryption (HE) and Functional Encryption (FE) that operate on encrypted data. Additionally, collaborative approaches involving distributed computations, such as secure Multi-Party Computation (MPC), statistical methods like Differential Privacy (DP), and hardware-based solutions utilizing Trusted Execution Environments (TEE), are presented to address privacy concerns in ML.
- In addition to addressing privacy concerns, we delve into the realm of verifiable Machine Learning (VML) in the PQ era. We explore techniques such as Zero-Knowledge Proofs (ZKP) and introduce quantum cryptography (QC). Furthermore, we discuss the next generation of ML-based intrusion detection systems (IDS), where ML plays a crucial role in detecting malicious attacks, serving as a network security countermeasure in PQ era.
- This article not only highlights gaps and implementation weaknesses in PQ-secure solutions that require further analysis but also presents PQ-

TABLE 1. A Qualitative Comparison Among PQ-Secure PPML Solutions

Approach	Advantages	Limitations	Utility
Homomorphic Encryption	-Data Privacy; Model Privacy -Function Privacy -Open-Source Library Implementation -Advanced Properties: -Attribute-based FHE, -threshold-FHE	-Heavy Computational Costs -Large Sizes -Low Performance efficiency -Limited Functionality for -Complex Models	-Training Phase -Inference Phase -Secure Aggregation -Centralized Learning -Distributed Setting
Functional Encryption	-Data Privacy; Model Privacy -Plaintext Output -Suitable for Inference Applications	-Heavy Computational Costs -Shortage of Open-Source Libraries -Requires Key Manager	-Training and Inference -MLaaS -Cloud-based ML
Multi Party Computation	-Low Computational Requirement -Data Privacy -Model Privacy -Permit Online-Offline Methods	-Heavy Communication Costs -Scalability Issues -High Bandwidth Requirements -Collusion Attacks	-Distributed Learning & FL -Untrusted Setting -Local Training -Private Aggregation
Differential Privacy	-Low Computational Requirements -Model Inversion Resistance -Inference Attack Resistance -Diverse Utility in ML beyond Privacy	-Low Accuracy -Honest-but-curious setting -Parameter Sensitive (ϵ) -Privacy Loss in repeated queries	-Centralized Learning -Distributed Setting -AI Applications: -Resampling, Model Testing
Trusted Execution Environment	-Data Privacy -Model, Software, Code Privacy -Lower Computational Overhead	-High Hardware Expenses -Potential Side-Channel Attacks -Extra Security Assumptions	-Confidential Computation -Private ML -FL with IoT Devices
Zero Knowledge Proof	-Complementary tool to PPML -Verifiability -Accuracy Assurance	-Not a PPML -Computationally Heavy -Proof Information Leakage -Impractical for Complex ML	-Verifiable Computation -Outsourcing Computations -Secure Aggregation -FL with Malicious Users
Quantum Cryptography	-Unconditional Security -Unauthorized Access Detection -Higher Accuracy -Faster Performance	-Still in its infancy -Implementation Challenges -Data Encoding -Expensive Infrastructure	-QML -QPPML -QFL -Intrusion Detection

secure visions, practical insights, and potential synergies of approaches as long-term safety measures along with a qualitative comparison among them as shown in Table 1.

A Provably Secure Approach to PPML

Homomorphic Encryption

Fundamentally, homomorphism is a property that arises from preserving the structure of a map. In the context of cryptographic encryption, it enables computations over encrypted data. HE can be classified into three classes based on the applied function: Partial HE (PHE), Somewhat HE (SHE), and Fully HE (FHE). The first proposal of an FHE scheme by Gentry [4] marked a pivotal moment, shifting its theoretical consideration as a holy grail of cryptography towards practical adoption in real-world applications. Technically, HE, especially FHE, is recognized as a privacy-preserving solution for Privacy-Enhancing Technologies (PETs) and plays a crucial role in almost all PPML scenarios. Figure 2 depicts the diverse classes, generations, and utility of HE in PQ-secure ML.

Due to substantial computational burdens, FHE-based approaches often allocate less to no emphasis on the training phase. As a result, the majority of state-of-the-art FHE-based PPML schemes primarily concentrate on the inference phase of ML, employing trained models in MLaaS for prediction over encrypted data. Nevertheless, there are researches that present methods for training neural networks over encrypted data without relying on unrealistic assumptions or distributed/client-assisted computations [5].

Several PHE and SHE schemes are rooted in classical cryptography and lack PQ promises. However, the majority of practical FHE schemes are indeed based on PQC, particularly formed on lattice-based cryptography, ensuring PQ security. The practicality of the HE approach is significantly influenced by FHE tools and existing open-source libraries. Numerous libraries incorporate state-of-the-art FHE schemes, with the aim of expanding their capabilities to support mathematical and optimization techniques. Advances in engineering aspects, including batching techniques for handling multiple inputs in a single ciphertext, support for parallelization and partitioning, as well as hardware acceleration through GPU and FPGA implementation, fast bootstrapping, and algorithmic enhancements for the training phase, contribute to increased efficiency and improved performance of FHE. These developments make FHE a viable solution for ML applications.

Many FHE schemes assume the encryption of all data under one key. A novel extension involves

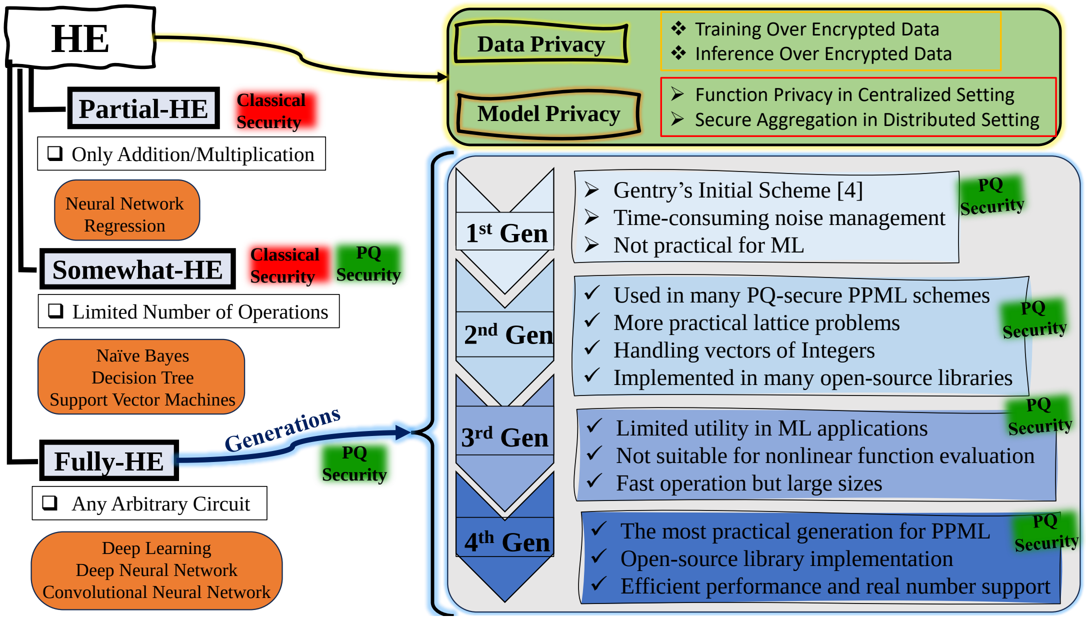
HE
Partial-HE
Classical Security
❑ Only Addition/Multiplication
Neural Network Regression
Somewhat-HE
Classical Security
PQ Security
❑ Limited Number of Operations
Naïve Bayes
Decision Tree
Support Vector Machines
Fully-HE
Generations
PQ Security
❑ Any Arbitrary Circuit
Deep Learning
Deep Neural Network
Convolutional Neural Network
Data Privacy
❖ Training Over Encrypted Data
❖ Inference Over Encrypted Data
Model Privacy
➢ Function Privacy in Centralized Setting
➢ Secure Aggregation in Distributed Setting
1st Gen
➢ Gentry's Initial Scheme [4]
➢ Time-consuming noise management
➢ Not practical for ML
PQ Security
2nd Gen
✓ Used in many PQ-secure PPML schemes
✓ More practical lattice problems
✓ Handling vectors of Integers
✓ Implemented in many open-source libraries
PQ Security
3rd Gen
✓ Limited utility in ML applications
✓ Not suitable for nonlinear function evaluation
✓ Fast operation but large sizes
PQ Security
4th Gen
✓ The most practical generation for PPML
✓ Open-source library implementation
✓ Efficient performance and real number support
PQ Security

FIGURE 2. Homomorphic Encryption Classes and Generations in PQ-Secure ML

expanding the number of keys that can be supported on homomorphically evaluated ciphertexts. While this extension poses challenges for efficient design, it is particularly suitable for PPML and AI-based applications in collaborative system models. This is especially relevant in scenarios involving distributed computation and FL, where data owners may lack trust or are unwilling to share a key. Another potentially valuable feature for PPML scenarios is distributed decryption enabling the combination of parties' secret keys to collectively decrypt the final ciphertext.

Incorporating Attribute/Identity-based encryption properties into FHE could render them suitable for ML applications like medical healthcare and financial detection. This enhancement not only preserve the privacy but also provides selective access control that allows only those who have a specific attribute or identity to access the encrypted data. Moreover, threshold-FHE enables the creation of a collective public key, allowing a group of users to participate in the evaluation process. Only a specific combination of these users can perform decryption. This property is well-suited for PPML in collaborative learning, encompassing both honest-majority and dishonest-majority scenarios.

The principle of function privacy, also known as circuit privacy in some contexts, within HE asserts that the confidentiality of the evaluated function/circuit must be safeguarded, revealing no additional information beyond the output result. This ensures that the output ciphertext appears indistinguishable from a freshly generated ciphertext. The integration of function privacy feature in the context of ML, could enhance the feasibility of utilizing FHE schemes, particularly in the training phase. Given that all these extensions are built upon existing FHE schemes, it can be asserted that they inherently ensure PQ security.

Despite its well-suitability for PPML scenarios, FHE comes with inherent limitations and challenges. The primary drawback is the substantial computational overhead and low performance efficiency, largely attributed to the lattice-based constructions employed. Furthermore, challenges such as large key and ciphertext sizes, the necessity for noise management, and limited functionality, particularly for complex neural networks over encrypted data, may constrain the utility of FHE in PPML applications. Despite efforts to develop practical FHE-based PPML designs, it is crucial to emphasize that, in comparison to tasks like inference, prediction, and classification over unencrypted data, there remains a considerable journey ahead to enhance output accuracy, reduce computational overheads for massive datasets, and alleviate performance burdens for real-world applications.

Functional Encryption

Functional Encryption (FE) can be seen as a rough generalization of a public key encryption. In FE, the holder of a specific secret key (i.e., functional encryption key) can compute a function on encrypted data without gaining extra information from the original plaintext or the master secret key. FE can be broadly classified into two types: Inner-Product Functional Encryption (IPFE) and Quadratic Functional Encryption (QFE), depending on the performed functionality. FE requires a trusted third party for key management to provide a decryption key based on the master secret key, allowing the key holder to execute a specific function on the encrypted data [6].

Comparable to employing an FHE scheme for PPML, FE is also utilized, with the key distinction being that the computation output is in plaintext format rather than ciphertext. Therefore, while both FHE and FE find applications in both training and inference phases of PPMML, FE may be more suitable for ML applications that necessitate plaintext output such as cloud-based ML and MLaaS. Besides, as opposed to HE-based approach, FE offers the advantageous feature of not requiring the sharing of the secret key for decrypting the computation output in ML inference.

While FE-based methods exhibit robust security, they necessitate a trusted third party for key generation, making them applicable only under the assumption of honest-but-curious security in PPML scenarios (i.e., the server fulfills its duties but seeks to obtain additional data whenever possible). It is essential to note that employing FE-based approaches for PPML is not bullet-proof and carries privacy risks. This risk arises from the plaintext format of the output result, introducing the potential for information leakage, and threats like Model Inversion and Direct Inference Attacks, even from an honest-but-curious server. FE-based PPML approach derives its security directly from the underlying employed FE scheme. However, the PQ assurances of FE-based PPML are contingent upon the difficulty of mathematical problems in the employed lattice-based or multivariate cryptography. These PQ guarantees entail efficiency sacrifices, including larger security parameters, slower operations, and increased complexities, making them practically unfeasible for large datasets.

As FE-based PPML is in its early stages, there are various challenges and opportunities that require attention and resolution:

- Currently, the FE-based schemes are more inclined towards the inference phase of ML. Thereby, there exists a gap in leveraging FE-based approaches for both training and inference phases over encrypted data, particularly those based on the QFE type.
- From practical aspects, FE-based approaches currently operate at comparable or lower levels than FHE-based PPML, incurring higher computational costs and exhibit limited functionality in managing complex functions, especially for intricate ML algorithms such as convolutional deep neural networks.
- In terms of implementation and its impact on performance, while FHE benefits from various libraries implementing multiple generations of its schemes across different programming languages and platforms, there are only a few libraries dedicated to implementing state-of-the-art FE schemes.
- From an engineering perspective, FE lacks attention, evident in the scarcity of hardware acceleration, GPU or FPGA utilization, and optimization methods for faster computational performance. Notably, to the best of our knowledge, all FE-based PPML implementations are currently on CPU, prompting the need for research on GPU support for FE schemes would significantly contribute to enhancing the applicability of FE-based PPML in real-world scenarios.

A Secure Collaboration Approach: Distributed Computation in PPFL

Secure Multi-Party Computation

As the name suggests, Multi-Party Computation (MPC) involves the secure distribution of function computation on private inputs among independent parties. This concept is particularly applicable in scenarios where a group of entities lacks mutual trust or when there is no trusted third party. The essence of collaborative computation in MPC lies in preserving privacy by ensuring the confidentiality of input data, correctness of output, and fairness. The novelty of MPC lies in its ability to almost entirely shift the computational costs away from individual parties, placing the burden on a powerful server. However, this comes at the expense of increased communication overhead among the involved parties [7].

Primarily, a secure MPC is implemented through cryptographic techniques, encompassing, among others, HE, Garbled Circuit (GC), Oblivious Transfer (OT), and Secret Sharing (SS). HE enables operations to be performed on encrypted data without decryption. OT allows for the exchange of private data in a privacy-preserving manner. SS enables data to be shared in a way that only a threshold combination of shares can recover the data, ensuring secure sharing. GC encrypts the boolean circuit version of the function to be computed, allowing for the secure evaluation of the function with no disclosure of intermediate and input data in a fixed number of communication rounds [8].

MPC enables distributed trust, giving rise to PPFL involving local data and model training while ensuring privacy. Applications of secure MPC techniques include collaborative model training with no data leakage, PPFL with private aggregation, outsourcing inference among distrustful servers, outsourcing the training phase of a model to independent servers, and even secure model/accuracy testing. This technique contributes to safeguarding the privacy of data and models in various distributed scenarios.

Indeed, MPC can be implemented using PQ-secure cryptographic primitives, such as NIST-standardized PQC or symmetric encryption schemes. Utilizing PQC-secure encryption during data preparation, employing a lattice-based secret sharing mechanism for data splitting, and incorporating PQ-secure HE schemes for aggregation/evaluation can provide PQ security assurances. Furthermore, MPC-based constructions that employ PQ-secure schemes under TEEs for function evaluation are resistant to PQ attacks, enhancing the overall security of the process.

The practical limitation of relying solely on MPC for PPML lies in scalability issues and substantial communication load, particularly in large-scale ML applications or training with massive databases. The performance challenges become more pronounced when PQ security measures are necessary. Additionally, the function must be known, public, or shared, which may not be feasible in certain scenarios. The potential for collusion among participants and the requirement for participants to be online with adequate bandwidth during function evaluation pose limitations on MPC-based approaches, particularly in ML applications with resource-constrained devices like IoT or mobile devices. An alternative approach for such scenarios involves using online-offline methods, allowing for precomputation and swift online interaction.

Given the computational power, communication bandwidth, and application choice, one may opt for FHE for low communication but high computation, or MPC for high communication with lower computation. Despite the longstanding competition between FHE and MPC as privacy solutions in various PETs, an optimal approach for PPML involves combining MPC and HE synergistically. This combination not only elevates privacy for both model and data owners, but it has also demonstrated significantly enhanced performance compared to using these techniques independently. However, it's worth noting that certain HE-enabled MPC techniques may lack resistance against model extraction attacks, requiring further exploration. One potential remedy to this issue is the integration of differential privacy on top of MPC. Finally, an advanced approach under exploration is multi-party quantum computation, a novel concept that requires further investigation.

A Statistical Approach towards PPML

Differential Privacy

Differential Privacy (DP) can be regarded as a technique employing distortion or perturbation mechanisms to anonymize query responses in a database. Its goal is to ensure that the presence/absence of a single data point is not revealed, with a privacy budget denoted by the factor ϵ. A smaller value of ϵ corresponds to increased anonymization or perturbation, thereby offering higher levels of privacy protection [9].

After extensive research spanning over a decade, DP has emerged as the predominant standard for privacy protection in PETs. In layman's terms, DP is a randomizing algorithm that incorporates calibrated statistical noise addition or carefully adjusted dataset swapping. In ML setting, the statistical features of the model output or inference remain preserved and are mathematically close to a regular output given a data record. This effectively prevents extraction attacks, such as model/data extraction, or inference attacks.

DP can be categorized based on architecture and learning setting. In centralized setting, given the sensitivity of the ML algorithm, the noise is added to the objective function. Its weakness lies in the honest-but-curious assumption and reliance on the model administrator, which may contradict privacy policies. In decentralized ML, local DP includes users adding noise to their data for privacy against the model, central DP blends fine-tuned noise with aggregated output to hide user activity during training, and instance-level DP introduces randomization to local labels to guard against the disclosure of final label information.

DP is commonly employed in conjunction with other techniques like MPC, HE, FE, or others to fortify PQ promises. An additional PQ dimension of DP involves the introduction of actual Quantum Noise, leading to Quantum Differential Privacy (QDP). In this context, quantum information processing can play a crucial role in ML, offering a viable solution to improve model and user data privacy in the PQ era. While QDP demonstrates resilience to post-processing attacks for any private datasets, the practical realization of quantum noise and algorithms for real-world applications poses significant challenges and inefficiencies. Consequently, introducing QDP to the concept of delegated quantum computation becomes a viable alternative. This involves delegating the computational aspect to a central server equipped with a large quantum computer, ensuring the protection of client-side computations while enabling verifiable outputs. This vision holds promise for the near future, potentially preceding the full realization of the PQ era.

Among PQ-secure approaches in PPML, avoids the costly performance overhead associated with encryption-based methods or the limitations seen in TEE-based approaches. Furthermore, DP can be applied to address common threats such as extraction, tracing, inversion, and reconstruction attacks. Beyond safeguarding privacy, DP offers diverse utilities in AI/ML applications. These include preventing over-fitting through DP-assisted data testing, ensuring model fairness by employing DP in data resampling, and addressing stability issues. Additionally, in multi-agent systems where AI acts as agents, DP can serve as a randomization tool to enhance communication and bolster security against malicious agents.

However, DP is not an all-encompassing solution. The intrinsic trade-off of privacy in DP lies in the compromise of output accuracy. Despite safeguarding individual identities through the censorship of personal data, DP diminishes the quality of the trained model, its parameters, or the output results.

PPML In Secure Environments: A Hardware-based Approach

Trusted Execution Environment

A Trusted Execution Environment (TEE), as the name suggests, establishes a secure, isolated, and trusted memory space for code execution. Access to data and code is protected through the utilization of both hardware and software assistance. Various TEE designs cater to different architectures, with Intel Software Guard Extensions (SGX) and ARM Trust Zone (TZ) being the most prevalent ones. In essence, a hardware-assisted approach like TEE promises the notion of "Confidential Computation". In the prevalent use of ML computations in untrusted or distributed environments, TEE guarantees the protection not only of the data in process but also safeguards the utilized software, codes, the entire ML model, ML computations, and memory registers within the TEE [10].

The use of TEE as a solution for achieving PPML is a relatively new approach. When applied to real-world ML applications, it doesn't ensure computational security levels and relies on higher assumptions but comprised of lower computational overhead compared to other cryptographic schemes based on mathematical hard problems (e.g., HE, FE, MPC). The PQ security of TEE-based approaches doesn't originate from the TEE design itself; instead, it relies on the utilization of a PQ-secure scheme under the TEE. Technically, by employing NIST PQC-standardized schemes for ML training and inference, ensuring that computations are not secure by design but by protocol, we can achieve PQ promises in PPML.

TEE can be employed either independently or in combination with other privacy solutions like DP, masking techniques, and encryption. This combination ensures different aspects of what is termed as secure and private ML. The deployment of TEE in distributed learning might be particularly beneficial, especially in applications involving edge-computing devices. However, it could necessitate additional requirements for coordination among different systems.

The TEE-based PPML faces challenges such as hardware expenses and susceptibility to side-channel attacks, including execution time and power analysis. These attacks could undermine the confidentiality of enclaves, posing a threat to the system by potentially leaking information and disclosing privacy. Additionally, in the context of ML with massive data or complex models, the necessity for partitioning and batch processing is essential, rendering TEE-based PPML inapplicable in some scenarios. Furthermore, the use of TEE is not optimal for performance. Additionally, the design of GPU support and acceleration techniques could prove beneficial for ML applications in the TEE context. Thus, there is a need for further studies and attention, particularly in the engineering aspect.

Beyond Privacy: Verifiable-PPML

Zero Knowledge Proof in ML

ZKP, at its core, a cryptographic two-party construction, seeks to safeguard data privacy as one party proves a statement to another without disclosing any additional information beyond the statement's validity, hence the term "zero-knowledge". The classification of ZKPs depends on the communication and interaction between the involved parties, leading to two categories: interactive and non-interactive protocols.

The presence of malicious parties in outsourced ML computations, such as FL involving user-assisted training or model aggregation, introduces potential vulnerabilities and risks. Specifically, the model owner might engage in dubious activities when applying the ML model, users could act maliciously or misbehave to gain advantage in the inference phase by poisoning input data, or they might input incorrect data to lower their computational costs, especially if they have limited resources. Also, in real-world AI applications involving sensitive information, such as in financial services (e.g., fraud detection, money laundering monitoring, trading), healthcare systems (e.g., AI diagnosis, medical insurance policies), etc., it is crucial for the model owner to correctly apply the model to users' data. Therefore, ensuring computational integrity (i.e., verifiability) alongside privacy becomes vital [11].

ZKP emerges as the ideal solution for these scenarios, fundamentally offering trust in the system, legitimizing computation correctness, proving the accuracy of model output, and ensuring training correctness in a privacy-preserving manner. Thereby, ZKP could be viewed as a complementary tool to other privacy-enhancing solutions for ML. Note that, despite the need for verifiable ML, the verification algorithm itself should not disclose users' privacy and resist against common threats like member inference and reconstruction attacks. While the majority of efficient ZKP systems are built on Elliptic Curve Cryptography (ECC) and offer short proofs with fast verification, there are also practically efficient ZKP schemes constructed on lattice-based cryptography and symmetric ciphers, providing PQ promises. PQ-secure ZKP, while not directly applicable in all situations where other PPML approaches are used, offers distinct advantages:

1) *vs. TEE*: ZKP can be viewed as an equivalent approach to those based on TEEs, where both protect computations and learning processes within a trusted environment. While TEEs rely on hardware assistance, ZKP provides cryptographic proof for secure computations. Despite differences in their fundamental assumptions and security assurances, ZKP incurs lower implementation costs.
2) *vs. Encryption*: While ZKP may not provide the same privacy guarantees for ML as HE or FE approaches, it certainly imposes less computational burden on the system.
3) *vs. MPC*: PQ-secure ZKP-based schemes are mostly non-interactive, in contrast to MPC-based PPML that assumes synchronous communication without support for dynamic parties. This results in a more realistic system model and lower overhead.
4) *vs. DP*: While a direct comparison may not be fair, the combination of these two could offer multiple levels of privacy protection for PPML.

An evident area for potential future work involves integrating VML into existing PPML frameworks, especially for those with PQ assurances. For example, in combination with encryption-based approaches, VML can validate that encrypted data is correct and falls within a specific range. Currently, they remain impractical for complex ML algorithms, can be extremely expensive for proof generation in many PPML scenarios with extensive tasks, and are limited in addressing specific aspects of privacy in ML.

Next-Generation Security In/By ML

Quantum Cryptography

Leveraging quantum mechanics for building cryptographic designs, the concept of quantum cryptography (QC) is introduced as a counterpart to PQC which relies on mathematically hard problems. The QC-based constructions employs physics characteristics such as interference, superposition, and entanglement to efficiently create and distribute secret keys and communication protocols with unconditional security guarantees. Achieving QC poses challenges, particularly in implementing fault-tolerant quantum computers with millions of qubits, which is still a work in progress and lacks practicality for real-world applications [12].

Integration of QC with ML gives rise to concepts such as QML, paving the way for advancements like Q-PPML, Q-FL, etc., elevating the efficacy of traditional ML algorithms. Irrespective of the learning paradigm, the performance benefits of QML over conventional ML are exponential, offering advantages such as increased accuracy, reduced loss, faster operational power through quantum parallelization, and enhanced security in the PQ era. Furthermore, its inherent principles enable the detection of any intrusion or unauthorized access, making QML particularly suitable for certain real-world applications.

QML research is categorized into three groups: 1) Pure QML utilizes quantum algorithms on quantum devices for learning, 2) Quantum-inspired ML enhances traditional ML algorithms using quantum computing, and 3) Hybrid quantum/classical ML combines algorithms from both worlds to minimize learning costs and enhance performance. In QML, there are four categories based on whether the data and algorithm are classical or quantum. Currently, the focus of research leans towards the practicality of applying quantum algorithms on classical data for real-world applications. While not a standalone privacy solution, QML can implement PPML, bolstered by PQC for enhanced privacy. Advancements can further enable Differentially Private QML. This underscores a significant research opportunity in the investigation of privacy aspects within the realm of QML.

ML-based Intrusion Detection Systems

The substantial increase in cyberattacks within networks highlights the need for advanced solutions beyond basic security measures. In these scenarios, ML serves as a crucial ally, functioning as our primary detection tool. Intrusion Detection Systems (IDS) are the deployed tools aimed at identifying any anomalous behavior in data flow to prevent and protect against malicious activities within the network. When an IDS scrutinizes the network using pre-defined flow patterns stored in a signature database to detect attacks, it is classified as a signature-based IDS. In contrast, an anomaly-based IDS employs a pre-defined model of normal behavior to identify abnormalities in network flow, labeling them as potential attacks. Thus, these two can be seen as complementary tools, collectively detecting both known and unknown attacks [13].

Thus, with advancements in quantum computing, the concept of a quantum IDS has emerged, with numerous research efforts concentrated on its realization. A pragmatic future direction in harnessing QML involves the adoption of hybrid classical/quantum algorithms for network attack detection. In this approach, the data and algorithms may be classical, while the methods (such as classification, neural networks, etc.) must incorporate quantum promises, such as quantum operations, to ensure efficacy in the PQ era. Despite the aforementioned discussions, the realm of secure QML is still in its nascent stages and demands substantial attention to address crucial concerns such as security and privacy preservation.

REFERENCES

1. M. Al-Rubaie and J. M. Chang, "Privacy-Preserving Machine Learning: Threats and Solutions," in IEEE Security & Privacy, vol. 17, no. 2, pp. 49-58, March-April 2019, doi: 10.1109/MSEC.2018.2888775.
2. F. Kerschbaum and N. Lukas, "Privacy-Preserving Machine Learning [Cryptography]," in IEEE Security & Privacy, vol. 21, no. 6, pp. 90-94, Nov.-Dec. 2023, doi: 10.1109/MSEC.2023.3315944.
3. S. Darzi, K. Ahmadi, S. Aghapour, A.A. Yavuz, and M.M. Kermani, "Envisioning the future of cyber security in post-quantum era: A survey on pq standardization, applications, challenges and opportunities". arXiv preprint arXiv:2310.12037.
4. C. Gentry, "A Fully Homomorphic Encryption Scheme," vol. 20, no. 9. Stanford, CA, USA: Stanford Univ., 2009
5. R> Podschwadt, D. Takabi, P. Hu, M. H. Rafiei, and Z. Cai. "A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption." IEEE Access 10 (2022): 117477-117500.
6. P. Panzade, and D. Takabi. "FENet: Privacy-preserving Neural Network Training with Functional Encryption." In Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics, pp. 33-43. 2023.
7. C. Weikeng, T. Hoang, J. Guajardo, and A. A. Yavuz. "Titanium: A metadata-hiding file-sharing system with malicious security." In Network and Distributed System Security (NDSS) Symposium. 2022.
8. A. Agarwal, J. Bartusek, V. Goyal, D. Khurana, and G. Malavolta. "Post-quantum multi-party computation." In Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40, pp. 435-464. Springer International Publishing, 2021.
9. A. E. Ouadrhiri and A. Abdelhadi. "Differential privacy for deep and federated learning: A survey." IEEE access 10 (2022): 22359-22380.
10. K.D. Duy, T. Noh, S. Huh, and H. Lee. "Confidential machine learning computation in untrusted environments: A systems security perspective." IEEE Access 9 (2021): 168656-168677.
11. C. Niu, F. Wu, S. Tang, S. Ma, and G. Chen. "Toward verifiable and privacy preserving machine learning prediction." IEEE Transactions on Dependable and Secure Computing 19, no. 3 (2020): 1703-1721.
12. A. A. Yavuz, S. E. Nouma, T. Hoang et al. "Distributed Cyber-infrastructures and Artificial Intelligence in Hybrid Post-Quantum Era." 2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA). IEEE, 2022.
13. G. Arnaldo, and M. Correia. "Towards quantum-enhanced machine learning for network intrusion detection." In 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), pp. 1-8. IEEE, 2020.

Saleh Darzi Saleh Darzi is a Ph.D. candidate in Computer Science and Engineering, actively engaged in research within the Applied Cryptography Research Laboratory at the University of South Florida. His primary research pursuits revolve around post-quantum and applied cryptography, with a focus on addressing challenges in the privacy and security of IoT, Blockchain technology, and network security. Saleh holds a Master of Science degree in Electrical Engineering (Communication-System) from K. N. Toosi University of Technology, Tehran, Iran, obtained in 2021. Prior to this, he earned his Bachelor of Science in Electrical Engineering (Electronic) from the Islamic Azad University of Central Tehran Branch, Tehran, Iran, in 2016. Contact him at salehdarzi@usf.edu.

Attila A. Yavuz Dr. Attila Altay Yavuz is currently an Associate Professor in the Department of Computer Science and Engineering (CSE) and the Director of Applied Cryptography Research Laboratory at the University of South Florida (USF). He was an Assistant Professor in the School of Electrical Engineering and Computer Science, Oregon State University (2014 - 2018) and in CSE USF (2018 - 2021). He was a member of the security and privacy research group at the Robert Bosch Research and Technology Center North America (2011-2014). He received his Ph.D. degree in Computer Science from North Carolina State University in August 2011. He is interested in the design, analysis, and application of cryptographic tools and protocols to enhance the security of computer networks and systems. He is a recipient of the NSF CAREER Award, Cisco Research Award (thrice - 2019,2020,2022), unrestricted research gifts from Robert Bosch (five times), USF Faculty Outstanding Research Achievement Award, USF Excellence in Innovation Award, and USF College of Engineering's Outstanding Research Achievement Award. His research on privacy-enhancing technologies and intravehicular network security is in the process of technology transfer with potential worldwide deployments. He has authored more than 95 products including research articles in top conferences, journals, and patents. He is a senior member of IEEE and a member of ACM. Contact him at attilaayavuz@usf.edu