Artificial Intelligence Security

White Paper

COGNIFOG

Emna Amri, Malvina Catalano,
Yohann Paulus, Yannick Roelvink

December 2024

CYSEC

1 Abstract

Artificial Intelligence (AI) is revolutionizing industries by enabling unprecedented levels of automation, decision-making, and innovation. From healthcare diagnostics and financial analytics to autonomous systems and critical infrastructure management, AI systems are at the heart of operations that directly impact lives and businesses. However, this growing dependency on AI brings a new set of challenges: how do we secure these systems and the valuable assets they rely on?

AI assets —training data, inference processes, algorithms and model parameters— are vulnerable to a wide range of threats, including data breaches, model tampering, adversarial attacks, and intellectual property theft. For example, tampered models in critical infrastructure systems could lead to catastrophic failures, while stolen AI models could cost organizations their competitive edge.

This white paper explores these challenges in depth, illustrating risks through real-world examples and providing solutions to protect AI assets. Technologies like confidential computing, federated learning, secure multi-party computation, and remote attestation are highlighted as essential tools in the fight against AI security threats. The paper also highlights CYSEC's expertise and product offerings, which are designed to secure AI systems throughout their lifecycle—from training and deployment to scaling across distributed environments. With solutions that leverage advanced security features such as trusted execution environments (TEEs), secure container orchestration, and attested launches, CYSEC provides robust tools and frameworks to ensure the secure deployment of AI systems across both cloud and edge environments.

Securing AI is not just a technical necessity; it's a call to action for businesses and innovators. AI systems hold immense potential, but their adoption must be paired with robust security measures to ensure trust, reliability, and resilience. Through a blend of technical and strategic insights, this white paper offers a roadmap for organizations seeking to protect their AI assets and maintain a competitive edge in a rapidly evolving landscape. By addressing these challenges head-on, we can ensure AI remains a powerful tool for progress, operating securely and reliably across applications.

Contents

1 Abstract **2**

2 Introduction **4**

3 Exploring Security Challenges in Applied AI **6**
- 3.1 Data Privacy Concerns . . . 6
  - 3.1.1 Scenario 1: Healthcare Diagnostics . . . 7
  - 3.1.2 Scenario 2: Credit Scoring Systems . . . 7
- 3.2 Adversarial Attacks . . . 7
  - 3.2.1 Scenario 1: Autonomous Vehicle . . . 8
  - 3.2.2 Scenario 2: Facial Recognition Systems . . . 8
  - 3.2.3 Scenario 3: Healthcare . . . 8
  - 3.2.4 Scenario 4: E-Commerce . . . 8
- 3.3 Model Inversion Attacks . . . 9
  - 3.3.1 Scenario 1: Facial Recognition Systems . . . 9
  - 3.3.2 Scenario 2: Healthcare Diagnostics . . . 9
- 3.4 Intellectual Property (IP) Theft . . . 10
  - 3.4.1 Scenario 1: Financial Trading . . . 10
  - 3.4.2 Scenario 2: Autonomous Vehicles . . . 10
- 3.5 Model Integrity and Authenticity in Remote Deployments . . . 11
  - 3.5.1 Scenario 1: Maintenance in Manufacturing . . . 11
  - 3.5.2 Scenario 2: Medical Imaging Diagnosis . . . 11
  - 3.5.3 Scenerio 3: Critical Infrastructure Alert Systems . . . 11

4 State of the Art: Tools and Solutions for AI Security **12**
- 4.1 Homomorphic Encryption . . . 12
- 4.2 Secure Multi-Party Computation . . . 13
- 4.3 Differential Privacy . . . 14
- 4.4 Federated Learning . . . 15
- 4.5 Confidential Computing . . . 16
- 4.6 Remote Attestation . . . 18
- 4.7 Adversarial Training . . . 19

5 CYSEC Products for AI Security **21**
- 5.1 ARCA Trusted OS . . . 21
  - 5.1.1 AI Isolation and Protection from Tampering . . . 21
  - 5.1.2 Orchestration of AI Workloads in Distributed Systems . . . 22
  - 5.1.3 Scalability of AI Systems . . . 23
- 5.2 Attested Launch of ARCA Trusted OS . . . 23
  - 5.2.1 Cloud-Based AI Model Hosting . . . 23
  - 5.2.2 Distributed AI Workloads Across Multiple Nodes . . . 24
  - 5.2.3 AI Model Deployment at the Edge . . . 24

6 Conclusion **25**

2 Introduction

Artificial Intelligence (AI) is transforming business operations in ways that seemed almost unimaginable a decade ago. Today, industries across the board are tapping into AI-driven solutions to make smarter decisions, automate complex processes, and solve problems faster than ever before. From healthcare to finance, retail to manufacturing, AI is reshaping traditional practices and giving organizations a valuable competitive edge.

As AI continues to become more integrated into everyday operations, it is fundamental to keep it secure. AI systems rely on huge amounts of data, often including sensitive or confidential information, which bring up serious concerns about privacy and security. This is especially important in fields like healthcare and finance, where the data used to train the AI models is particularly sensitive, involving personal information, medical records, financial transactions, and many other confidential information.

On top of that, companies are investing heavily in developing AI models, and protecting them from misuse or theft is critical to maintaining a competitive edge and safeguarding Intellectual Property (IP). As governments and industries introduce stricter data protection and cybersecurity regulations, securing AI systems also helps organizations stay compliant.

To fully understand the scope of security AI, it is important to identify the key assets at risk in an AI context:

- **Models and algorithms** - These are the core intellectual property of an AI system, embodying years of research, development, and financial investment. They consist of the mathematical frameworks, processing methodologies, and decision-making logic that underlie AI capabilities. Models and algorithms drive the company's competitive advantage, differentiating its offerings and unique operational value.
- **Configuration and optimization parameters** - These are specific, fine-tuned settings that control the functionality, accuracy, and efficiency of an AI system in real-world operations. They include hyperparameters, threshold values, and tuning adjustments that are critical for maximizing performance and reliability.
- **Training data** - The dataset used to teach an AI model to recognize patterns and make accurate predictions. It serves as the model's learning material, often containing proprietary, sensitive, or regulated information.
- **Inference data** - This refers to the live, real-time data that an AI system processes to generate insights and make operational decisions. Figure 1 illustrates the difference between training and inference in AI, where training involves learning from a large dataset with error correction through backpropagation, while inference uses this learned knowledge to make predictions on new data without further adjustments.

Therefore, securing AI models and the data used to train those models is essential to ensure the integrity, confidentiality and trustworthiness of the systems that organizations and individuals increasingly rely on. The next chapter explores the security challenges posed by AI, also providing real-world examples for each challenge.

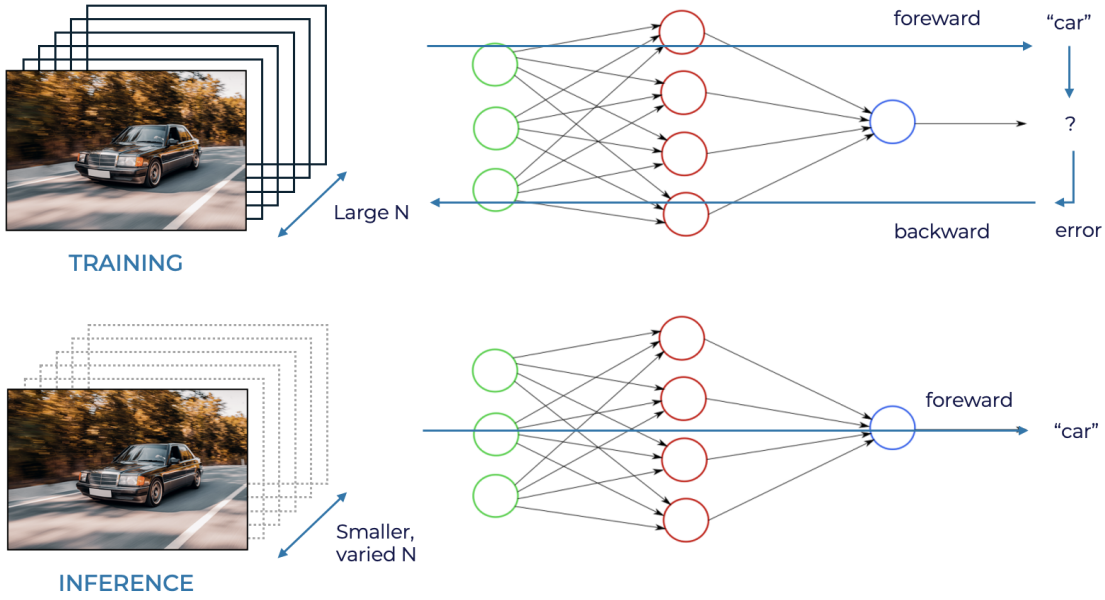

Figure 1: Training vs Inference

3 Exploring Security Challenges in Applied AI

The increasing dependency on AI models does not come without risks. As organizations leverage AI to automate decisions, enhance efficiency, and unlock valuable insights, they also expose themselves to **a new landscape of security vulnerabilities**. AI models now play critical roles in sectors like finance, healthcare, autonomous systems, and beyond, handling sensitive data and making influential decisions. This elevated importance makes them attractive targets for malicious actors aiming to exploit vulnerabilities for financial gain, privacy breaches, or to disrupt critical infrastructure.

In this section, we explore the security challenges that come with the deployment and usage of AI, from data privacy concerns to model tampering and adversarial attacks. **By examining these risks in real-world contexts, we can better understand the protective measures needed to build resilient, trustworthy AI solutions.** Figure 2 provides an overview of the main security risks associated with AI, each of which will be discussed in the following sections.

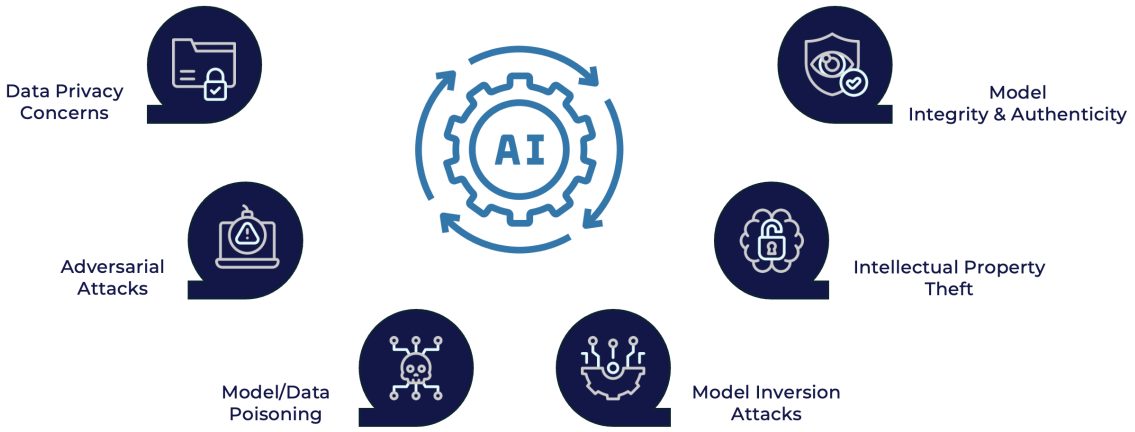

Figure 2: Security Risks Posed by AI

3.1 Data Privacy Concerns

Within the context of AI applications, data privacy concerns arise in two main are:

- **Training data**: AI models require enormous amounts of data, in order to cover the full range of queries the model in question will receive in operations. As an example, it is estimated that ChatGPT 4.0 was trained on around 300 Billion words, equal to roughly 570 GB of data, with various different sources (e.g. Wikipedia, research articles, webtexts, websites, and other forms of content) [1]. While confidentiality is less of a concern for models trained on public data, it becomes critical for models trained on sensitive information, such as patient records in the medical field. In such cases, maintaining strict data privacy is essential to protect the confidentiality of the individuals represented.

- **Inference data**: Once trained, the model is deployed to perform tasks in real-world environments, where it draws conclusions from new inputs, a process known as inference. In cases where user data, such as prompts in generative AI models, is part of the inference process, additional privacy protections are required. This includes protection against data exposure risks from potentially compromised hardware, as well as addressing privacy vulnerabilities within the model itself. Some implementations may incorporate inference data to further refine the model, potentially introducing user input data into subsequent training. To prevent data leaks during future inferences, these implementations must ensure that any personal or confidential inference data is securely excluded from the training dataset. For instance, ChatGPT includes this option, and users are encouraged to deactivate it in the settings for enhanced privacy and security.

Data privacy concerns for AI are highlighted in **Healthcare and Finance** as illustrated in the following use cases.

3.1.1 Scenario 1: Healthcare Diagnostics

In the healthcare sector, one of the main concerns when adopting new technologies is data privacy, given the sensitivity of patient data. Although healthcare organizations are beginning to adopt AI to improve patient care, improve diagnostics, and accelerate medical research, they remain deeply focused on maintaining data privacy and security. According to the Zscaler 2024 AI Security Report [5], healthcare accounts for only 5% of AI/ML traffic in the Zscaler cloud, with 17.23% of this traffic blocked to ensure security. This cautious approach shows how healthcare organizations prioritize data protection over rapid innovation.

An example of data privacy concerns in AI-driven healthcare diagnostics, where models are trained on sensitive patient data to predict health outcomes. If this data is improperly secured or anonymized, there is a risk of exposing Personal Identifiable Information (PII) during model training, sharing, or deployment. Such exposure can violate privacy regulations like HIPAA in the USA or GDPR in Europe, resulting in heavy fines and legal repercussions. Beyond regulatory issues, breaches in patient data privacy weaken trust in AI applications in healthcare, deterring patients from engaging with AI-driven services and damaging the institution's reputation.

3.1.2 Scenario 2: Credit Scoring Systems

Another example of data privacy concerns is seen in AI-based credit scoring systems used by financial institutions to assess loan eligibility. These models are often trained on sensitive personal information, including credit history, income levels, and spending patterns, to predict an individual's creditworthiness. If the model or data are not protected, there is a risk that unauthorized parties could access this private information, potentially exposing individuals' financial details. Similarly to the previous use case, such a breach would violate privacy regulations, like the GDPR or the Fair Credit Reporting Act (FCRA) in the USA, and also harm customer trust, as people rely on banks to protect their financial data.

3.2 Adversarial Attacks

Adversarial attacks involve subtly modifying input data to mislead an AI model. These modifications exploit the model's sensitivity to minor variations, causing incorrect classifications or decisions. There are several types of adversarial attacks, depending on the attacker's objectives, the model's vulnerabilities and the stage of the AI life-cycle being targeted (training, inference or deployment).

Figure 3 illustrates the various types and stages of adversarial attacks that target AI models, highlighting both offensive and defensive actions throughout the AI lifecycle. The **attacker's tactics** are shown in red and include techniques like **data poisoning**, where malicious data is injected into the training set to distort model predictions, **evasion**, where crafted inputs are used to trick the model during inference, and **extraction**, which involves extracting sensitive information or recreating the model's functionality by querying it. On the other hand, **defensive measures** are represented in blue. These include **adversarial training**, where the model is pre-trained on adversarial examples to enhance its robustness, **evasion detection**, which monitors for suspicious inputs that could indicate evasion attempts, and **poison detection** which safeguards the training data by identifying potentially poisoned data points. Certification and verification are also used to verify the model's integrity and ensure its outputs remain trustworthy across different environments.

Adversarial attacks pose significant risks, especially in fields where accuracy is essential, like **security, healthcare and autonomous systems**. In particular, data poisoning attacks can be difficult to detect, as the model's errors may appear as natural performance fluctuations rather than intentional tampering. Therefore, these attacks have serious implications for fields that depend heavily on data integrity, such as healthcare, finance, and e-commerce, where they can compromise service quality, introduce safety risks, and damage trust in AI-driven tools. Some examples are described here.

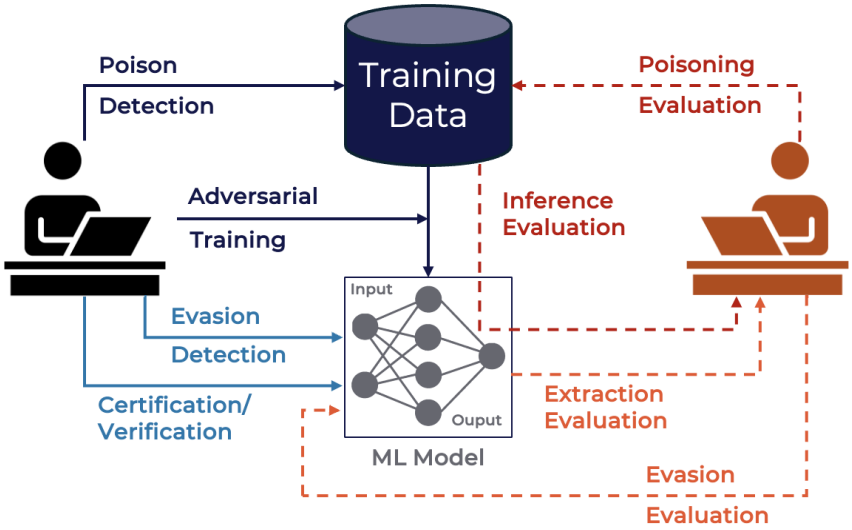

Figure 3: Adversarial Attacks Overview

3.2.1 Scenario 1: Autonomous Vehicle

An example of an adversarial attack is in autonomous vehicle vision systems, where small, almost imperceptible changes to road signs or objects can mislead the vehicle's AI. For instance, by applying stickers or paint to a stop sign, an attacker can cause the AI to misclassify it as a speed limit sign or other objects, leading the vehicle to ignore it entirely. This misclassification poses serious safety risks, potentially resulting in traffic violations or accidents, as the vehicle may fail to stop at intersections. Such attacks highlight the vulnerabilities in AI perception systems and the need for robust defenses in safety-critical applications like autonomous driving. It is also one of the main reasons why the adoption of autonomous driving has been slow.

3.2.2 Scenario 2: Facial Recognition Systems

In facial recognition, small changes, like patterns on accessories, can trick the AI into misidentifying individuals. This can enable unauthorized access or allow individuals to evade surveillance. For example, if a bank's mobile app uses facial recognition to authenticate payments, an attacker could use these subtle alterations to trick the system into authorizing a transfer as if they were the legitimate account holder. This type of vulnerability exposes significant security risks, as it bypasses a major layer of identity verification.

3.2.3 Scenario 3: Healthcare

In the healthcare sector, where AI models are used for diagnostics, predictive analytics, and patient care recommendations, the impact of data poisoning can be particularly severe. For instance, an attacker could alter patient records by introducing false symptoms or inaccurate diagnostic information, which could then alter diagnostic algorithms. This might lead the AI system to misdiagnose patients or recommend incorrect treatments, potentially harming patients' health. Over time, these errors could have a domino effect, as each incorrect diagnosis may influence future model updates, amplifying inaccuracies and reducing overall effectiveness in clinical settings.

3.2.4 Scenario 4: E-Commerce

In e-commerce, product recommendation systems are crucial for enhancing sales and customer experience, relying heavily on user interaction data to refine recommendations. In a data poisoning attack, a malicious competitor or insider could inject misleading data into this system, subtly manipulating user interactions

(e.g., fake clicks, purchases) to bias recommendations in their favor. Over time, these poisoned data points shift the algorithm's perception of relevance, altering recommendations to promote specific items while impacting other vendors' sales. This affects customer trust and satisfaction, as users see increasingly irrelevant recommendations, potentially driving them to competing platforms. In addition, these attacks could lead to revenue loss, costly system audits, and reputational damage if exposed.

3.3 Model Inversion Attacks

Model inversion attacks occur when an adversary leverages access to an AI model to reverse-engineer and reconstruct sensitive data from the training set. As shown in Figure 4, during the **training phase**, a model is fed a dataset to learn specific patterns and features. Once deployed as an ML-based service in the **testing phase**, attackers can exploit a prediction API to query the model. By exploiting the patterns learned by the model, attackers can extract private or confidential information that was intended to remain protected. This risk is particularly significant for models handling highly sensitive information, such as personal data, biometric information, or proprietary knowledge. Model inversion attacks present a serious challenge because, unlike direct data breaches, they exploit the very mechanisms that allow AI models to generalize from data, using indirect inference to access private information.

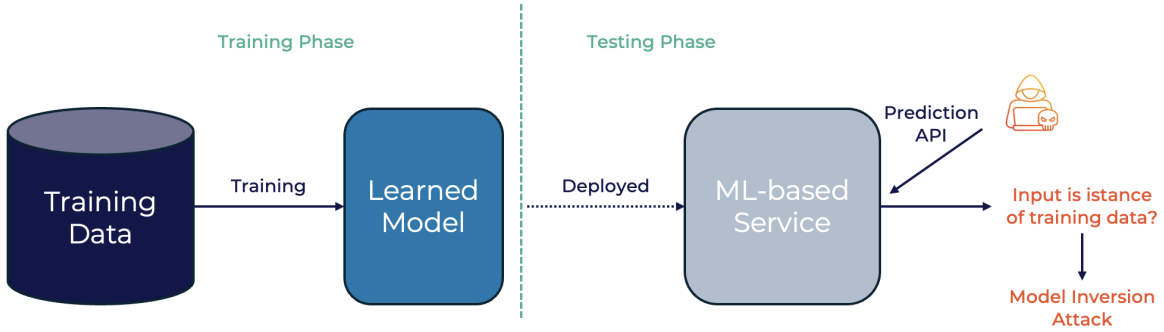

Figure 4: Model Inversion Attack Overview

The impact of these attacks extends beyond simple data leaks, as they can expose data previously thought to be anonymized or safely embedded in the model. This vulnerability is particularly concerning in fields where confidentiality is essential, such as **healthcare, law enforcement, and finance,** where breaches can have ethical, legal, and reputational repercussions. More details are provided in the following scenarios.

3.3.1 Scenario 1: Facial Recognition Systems

In facial recognition systems, model inversion attacks are a major concern. Attackers can use model inversion techniques to reconstruct facial images of individuals included in the model's training set. In practice, this means that if an adversary gains access to the model, they could generate approximate images of individuals whose data was used to train the system. Such an attack carries severe privacy implications, especially in sectors like law enforcement and public safety, where facial recognition models are commonly deployed.
For example, a law enforcement agency might use a facial recognition model trained on a database of surveillance footage. If an attacker manages to access this model, they could potentially reconstruct facial features of individuals within the database, violating both individual privacy and law enforcement protocols.

3.3.2 Scenario 2: Healthcare Diagnostics

As mentioned before, the healthcare sector faces many challenges of data privacy due to the sensitive nature of patient data in addition to model inversion attacks, being a critical risk that must be carefully mitigated. For example, if an attacker gains partial access to a healthcare diagnostic model, they might use inversion techniques to extract specific details about individual patients within the training dataset. This could include reconstructing medical conditions, extracting past treatments or even recreating diagnostic images. This can

result in exposure of patient private health information and put healthcare providers at risk of violating privacy regulations like HIPAA, GDPR and similar frameworks worldwide.

3.4 Intellectual Property (IP) Theft

The theft of AI algorithms and models represent one of the most significant security risks in applied AI. These models often include proprietary knowledge, years of research and substantial investment, making them highly valuable assets that define a company's competitive advantage. Protecting the IP of AI models is critical, as the loss or unauthorized replication of these assets can weaken a company's market position, lead to financial losses and compromise sensitive applications that rely on these systems.
When AI models are deployed, especially in third-party environments like cloud platform, edge devices or shared infrastructures, they become vulnerable to theft. Adversaries can recreate the model and bypass the costly research and development by extracting the model parameters and logic. This risk is amplified in industries where proprietary algorithms drive innovation and competitiveness, like finance, autonomous systems and manufacturing. The main security vulnerabilities AI models face include:

- **Parameter extraction attacks**: these occur when adversaries exploit access to a model (e.g., through APIs or endpoints) to extract weights and biases, which are essential to the model's functionality.
- **Reverse engineering**: attackers can de-compile AI applications to understand and replicate proprietary algorithms
- **Insider threat**: employees or collaborators with access to sensitive model files or training environments may leak or sell the proprietary information.
- **Deployment vulnerabilities**: models deployed in insecure environments can be intercepted or tampered with, exposing valuable assets.

It is not only the loss of competitive advantage that makes IP theft in AI so critical, stolen models could also be misused, resulting in reputational damage or legal liabilities for the affected organization.

In this context, initiatives like the **EU AI Pact** [2] provide support to organizations as they navigate the challenges of AI and IP protection. The pact encourages companies to voluntarily implement the AI Act's requirements ahead of its full application. Over 100 companies, ranging from multinational corporations to SMEs have signed the pact, committing to key actions to develop an AI governance strategy and mapping high-risk AI systems. This pact is fundamental to help companies proactively **address IP protection, data privacy and security** concerns while ensuring that their AI systems align with the evolving regulatory landscape.

3.4.1 Scenario 1: Financial Trading

A common example of IP theft occurs with proprietary AI models in financial trading. Financial firms invest heavily in developing AI models that analyze market data, predict trends, and inform trading strategies. If a competitor or insider gains unauthorized access to one of these models, they can replicate the model's decision-making framework without the substantial R&D investment. This incident would enable the competitor to exploit similar trading strategies, potentially leading to financial losses for the original firm.

3.4.2 Scenario 2: Autonomous Vehicles

The autonomous vehicle (AV) industry is another area where IP theft represents significant risks. AV companies invest billions in developing sophisticated AI models that process sensor data, navigate environments, and make real-time decisions. These models include proprietary algorithms for perception, decision-making, and control, forming the core competitive advantage of each company. If an adversary gains access to these models, whether through cyber attacks on connected vehicles, insider threats, or breaches in third-party manufacturing facilities, they can replicate or reverse-engineer these proprietary systems. Such a breach would jeopardize the victim company's market position and raise safety concerns if the stolen model is deployed without proper safeguards. Moreover, unauthorized use of stolen IP could lead to regulatory challenges, as cloned systems may fail to meet industry standards, causing reputational damage to the original developer.

3.5 Model Integrity and Authenticity in Remote Deployments

Deploying AI models to remote infrastructure, such as cloud environments or edge devices, introduces significant risks related to model integrity and authenticity. These models, developed through extensive training involving billions of parameters and hyperparameters, are highly valuable assets that require protection from unauthorized modifications. During training, parameters like model weights and biases are fine-tuned to achieve optimal performance, while hyperparameters control the overall training process itself. For large models, such as OpenAI's GPT-3 with approximately 175 Billion parameters, this training process is time-intensive and resource-heavy, making the trained model parameters critical proprietary assets. Therefore, any unauthorized access to or tampering with these parameters represents both a security risk and a potential economic loss. This risk is particularly concerning in high-stakes applications like **financial forecasting, healthcare diagnostics, and autonomous vehicles**, where the model's accuracy and reliability directly impact safety and user trust.

3.5.1 Scenario 1: Maintenance in Manufacturing

In manufacturing, AI models are frequently used for predictive maintenance; analyzing machine data to predict equipment failures before they occur. These models, deployed on remote industrial IoT (Internet of Things) devices or cloud platforms connected to factory equipment, help reduce downtime and optimize maintenance schedules. However, deploying these predictive models on remote infrastructure introduces the risk of tampering, where an attacker could modify the model's parameters to misreport machine health, either by underestimating faults or falsely flagging issues. For instance, tampering with model parameters could cause critical equipment faults to go undetected, leading to unexpected equipment failures and costly downtime. Alternatively, false positives could prompt unnecessary maintenance actions, wasting resources and disrupting production.

3.5.2 Scenario 2: Medical Imaging Diagnosis

As previously mentioned, in the healthcare sector, AI models are increasingly used in diagnostic imaging, such as identifying tumors in radiology scans. These diagnostic models are often deployed on cloud servers or edge devices within medical facilities to process sensitive patient data locally. However, remote deployment exposes these models to tampering risks, where malicious actors could alter the model's parameters or decision-making logic. For example, an attacker could reduce the model's sensitivity to certain types of tumors, potentially leading to missed diagnoses, threatening patient health, and exposing healthcare providers to significant legal and reputational risks. However, the tampering risk can extend to patient privacy, as attackers could manipulate model outputs to reveal sensitive information embedded in the model or outputs.

3.5.3 Scenerio 3: Critical Infrastructure Alert Systems

AI systems are deployed extensively to process alerts and manage operations for critical infrastructure across distributed environments, such as power grids, water supply networks, and public transportation systems. These AI models analyze real-time data from sensors and edge devices to detect anomalies, predict failures, and issue timely alerts to prevent disruptions. Given the decentralized nature of these deployments, the models are often deployed on remote edge devices exposing them to significant risks of tampering. For instance, an attacker could manipulate the model's parameters or inject malicious code to compromise the integrity of the alerts, either suppressing critical warnings or generating false positives to disrupt operations. Such tampering could lead to catastrophic consequences, such as undetected power outages, compromised water quality, or large-scale transportation failures. The high stakes of these scenarios make it imperative to ensure model integrity and authenticity through robust security mechanisms, as even minor compromises can undermine safety, affect public trust, and cause severe economic losses.

The security challenges surrounding AI applications are multifaceted and span across several areas, from data privacy to model integrity, adversarial manipulation, and intellectual property protection. As organizations increasingly depend on AI models across various domains, these challenges highlight the need for robust security measures to protect data, models and trust in AI-driven solutions. To address these challenges, tools and methodologies are emerging to enhance AI security. The following chapter explores the state-of-the-art solutions currently available, examining tools and methodologies designed to safeguard AI assets, ensure data privacy, and improve model resilience against threats.

4 State of the Art: Tools and Solutions for AI Security

From protecting sensitive training data to ensuring the integrity of deployed models, organizations face complex challenges in maintaining the security and privacy of their AI assets. Fortunately, advancements in security technologies are providing robust solutions tailored to the unique vulnerabilities of AI systems. This chapter explores the available tools, frameworks, and approaches currently shaping the landscape of AI security.

4.1 Homomorphic Encryption

Homomorphic Encryption (HE) encompasses several novel encryption methods that allow for computations to be performed directly on the encrypted data, without the need to decrypt the underlying information. As such, it offers a robust security implementation for AI assets and implementations, as it limits the exposure of the sensitive training and inference data used within the models.

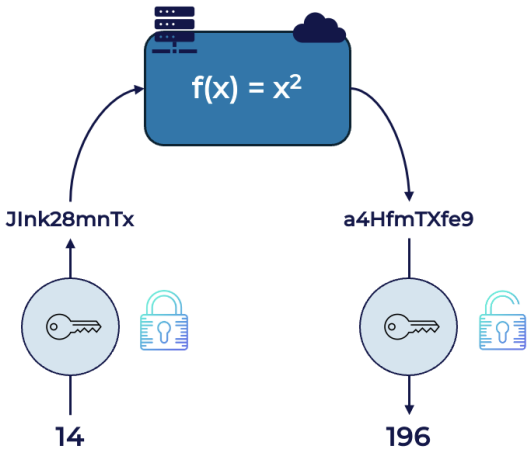

Figure 5: Illustrative example of Homomorphic Encryption

Although the field of HE is still an active field of research, several applications of varying degrees of maturity currently exist. One notable application is **Concrete ML**, an open-source library developed by Zama [26]. It enables data scientists to convert existing Machine Learning (ML) models into their homomorphic equivalents, facilitating privacy-preserving computations on encrypted data without requiring expertise in cryptography. Other noteworthy mentions are IBM's implementations for Full Homomorphic Encryption (FHE) for AI and machine learning, called **HElayers** and **HE4Cloud** [10] [22]. HELayers is a software development kit (SDK) that facilitates the development and serving of ML models using HE, while HE4Cloud is a beta cloud service that leverages the HELayers SDK for cloud-based ML deployments.

While HE offers strong privacy protection for AI assets by allowing computations on encrypted data, it comes with several key disadvantages:

- **Data size expansion:** The encryption of plaintext data within HE significantly increases the total size of the data set, ranging up to tens of thousands of times the size of its original, unencrypted version. Although research has been performed to reduce this radical increase [21], no general solution has been found yet.
- **Computational & time overhead:** Computational operations on the (expanded) encrypted data sets are inherently more intensive, requiring specialized and powerful hardware, and take substantially longer than similar operations on the equivalent plaintext data.

- **Noise impact & accuracy reduction:** HE introduces noise into encrypted data, which can impair an AI models' accuracy and performance. To mitigate this concern, AI systems often use approximation functions (e.g., ReLU) within the utilized polynomial expressions, which can further reduce precision.
- **Limited compatibility with complex models:** While HE is effective for simple models or specific types of machine learning tasks, it struggles with the computational demands of deep learning models due to the heavy processing requirements. This limits its applicability in complex AI tasks that require deep neural networks.

The aforementioned issues, among others, will have to be resolved before FHE can be utilized to secure AI implementations on a large, industrial scale.

4.2 Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) enables multiple parties to collaboratively compute functions over their inputs while keeping those inputs private. It achieves this by splitting each party's data into encrypted portions, which are distributed across the computational nodes of the SMPC implementation. These nodes then perform the necessary calculations using only the shares of data they are provided, without accessing the complete dataset. When completed, the results from each of the nodes are gathered and combined to provide the final results. An overview of a typical implementation is provided in Figure 6 [19].

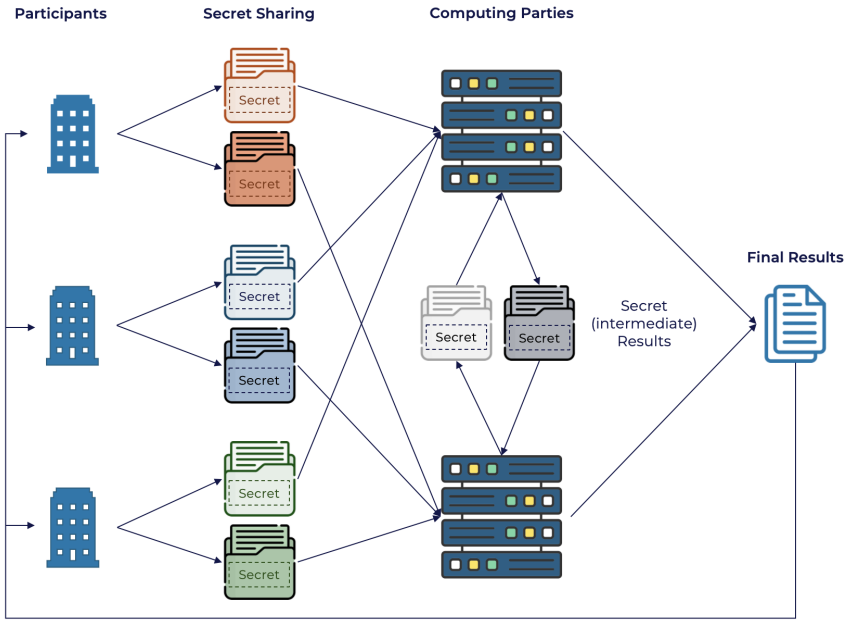

Figure 6: An overview of an SMPC implementation leveraging the data from two separate parties

The system ensures that any individual node or party lacks the information required to reconstruct the original inputs, thus safeguarding privacy. In addition, SMPC can be combined with HE schemes, in order to add additional security against malicious nodes.

Notable SMPC implementations include Meta's **CrypTen** [20], an open-source framework that integrates SMPC with machine learning, allowing for the training and evaluation of models on encrypted data without exposing sensitive information, and **Duality Technologies' Secure Data Collaboration Platform** [3], which leverages a combination of HE, SMPC and other Privacy Enhancing Technologies (PETs) to allow its users to perform joint computations and analytics on encrypted data.

However, SMPC does have several limitations that impact its practical deployment, especially in resource-intensive AI applications:

- **Computational overhead:** SMPC protocols require significant computation and data exchange between participating parties, arising from the cryptographic processes needed to maintain privacy, like secret-sharing and encryption. As a result, SMPC can be slower and more resource-intensive than traditional computations, which can make it impractical for real-time applications or for large datasets common in AI.
- **Communication & Scalability challenges:** SMPC often requires extensive inter-party communication, as participants must frequently exchange shares of intermediate results to ensure the final computation remains secure. As the number of these communications grows exponentially with the number of participants, this puts a hard limit on the feasibility of large-scale SMPC-based AI implementations.
- **Limited functionality for deep learning:** While SMPC works well for simpler machine learning models and linear computations, it struggles with the high computational demands of deep learning. Complex neural networks, which involve non-linear operations and extensive matrix multiplications, face severe performance bottlenecks when implemented with SMPC, making it challenging to apply SMPC to large-scale deep learning tasks.
- **Lack of standardization & interoperability:** SMPC frameworks and protocols are diverse, and standards are still emerging. This lack of interoperability between different SMPC implementations makes it challenging to integrate with existing AI frameworks and limits the potential for cross-industry collaborations.

4.3 Differential Privacy

Differential Privacy (DP) is a statistical technique that ensures individual data points cannot be reverse-engineered from aggregated results, e.g. the computational results within an SMPC architecture [17]. It achieves this by introducing mathematical noise to the dataset or computations so that the influence of any single data point becomes indistinguishable, thus ensuring individual privacy even if the model or data is exposed. As such, DP is widely used within AI model implementations, enhancing the privacy of the utilized training and/or user input data.

In general, two distinct types of DP are defined, called **Local** and **Global**. Where Local DP introduces noise to the *individual data points* before they are aggregated, Global DP adds noise to the *output data* of the aggregation process. An illustration of the differences between these two processes is provided in Figure 7 [16].

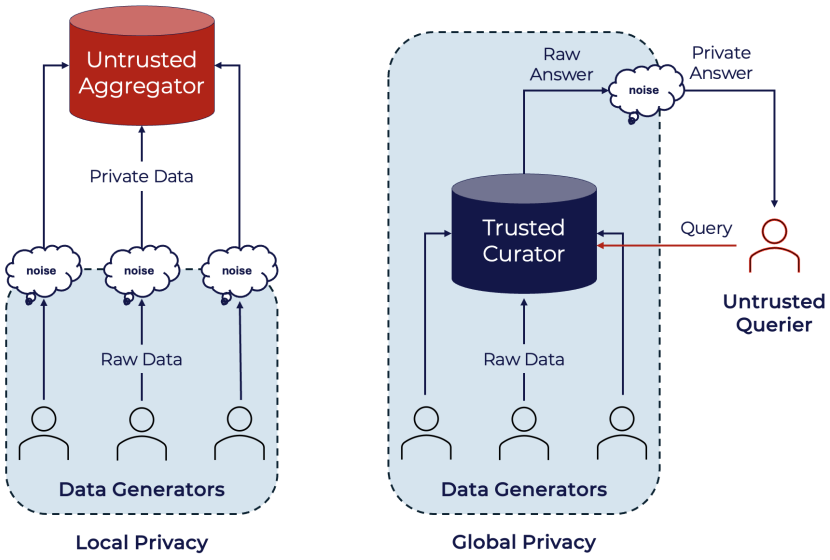

Figure 7: An illustration of a Local (left) and Global (right) DP implementation

Several open-source implementations for DP exist, with the most noteworthy ones being **OpenDP**, developed by Harvard's Institute for Quantitative Social Science (IQSS) and School of Engineering and Applied Sciences (SEAS) [7], **Tumult Analytics**, developed by Tumult Labs [14], **PipelineDP**, co-developed by Google and Openminded [4], and **Diffprivlib**, developed by IBM [6].

However, there are several limitations and trade-offs linked to the usage of DP:

- **Data utility reduction:** The addition of noise to AI model datasets or computations can lead to significant reductions in accuracy. This is particularly challenging for tasks requiring high precision or granular insights, as excessive noise can render the data less useful or distort results.
- **Privacy budget depletion:** DP operates on a *privacy budget* concept, meaning that each query or model interaction consumes part of the allotted privacy budget [25]. In dynamic systems with ongoing queries or continual data updates, the privacy budget can be exhausted quickly, limiting the system's ability to deliver useful insights while maintaining DP guarantees.
- **Parameter tuning and maintenance:** The parameters that define the DP's privacy budget and noise levels need to be carefully selected, and are highly context-depended. As such, the tuning and management of these parameters is an ongoing and resource-hungry process throughout the model's life-cycle, limiting its applicability to large datasets or complex models.

4.4 Federated Learning

Federated learning (FL) is a decentralized approach to training machine learning models across multiple computational nodes without needing to aggregate the data on a central server. It achieves this by allowing each individual node to train on their local data, and only periodically share their proposed module updates to a central aggregator. The central aggregator is then able to construct a global model that was trained using several private data sets, while maintaining the privacy of each of them. It contrasts with SMPC as only the module updates are shared with the final aggregator, rather than subsections of the training data. FL is particularly useful for AI applications where data privacy is critical, such as healthcare and finance. An example implementation of FL is provided in Figure 8.

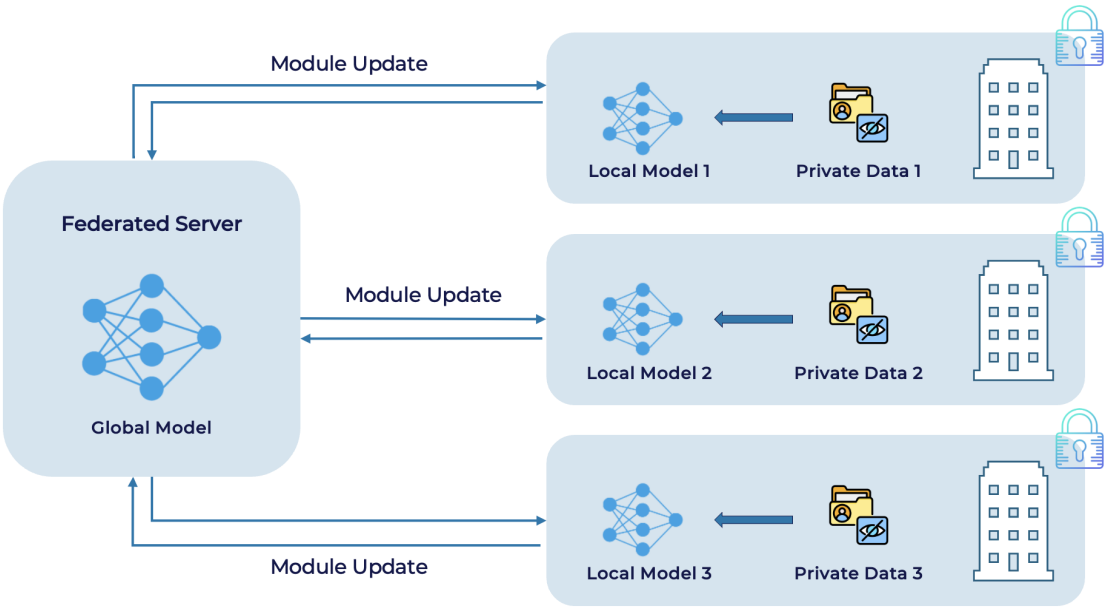

Figure 8: Illustration of an AI model being trained on private data using Federated Learning

A wide range of existing open-source FL implementations exists, with the most notable ones being **NVFlare** developed by NVIDIA [23], **TensorFlow Federated** developed by Google [13], **OpenFL** developed by Intel

[9], and **PySyft** developed by OpenMinded [8].

While FL is able to significantly enhance the privacy of a model's training data, it is not perfect, and still suffers from several limitations:

- **Limited security:** Although FL networks only expose module updates to a centralized server, it is nonetheless possible to extract sensitive data from this information. To combat this, FL is regularly combined with other privacy enhancing technologies, such as DP and SMPC, which will eventually increase the globlal computational overhead.
- **Statistical data heterogeneity:** FL networks often involve non-Independent and Identically Distributed (non-IID) data across participants, meaning that each client's data may follow different distributions. This data heterogeneity can lead to slower model convergence and poor generalization, as models trained on one client's data may not perform well on another.
- **Communication & Scalability challenges:** Similar to SMPC, FL requires extensive inter-party communication, which scales exponentially with the number of participating devices.

4.5 Confidential Computing

Another widely used concept used to protect AI model data is Confidential Computing (CC). Where regular encryption methods can be used to secure data while it is at rest (e.g. when it is stored in a database) or in transit (e.g. when it is transferred to the network), CC aims to protect data while it is in use (e.g. when data is being processed). For AI implementations, which often have to process sensitive information, this is crucial for maintaining data privacy.

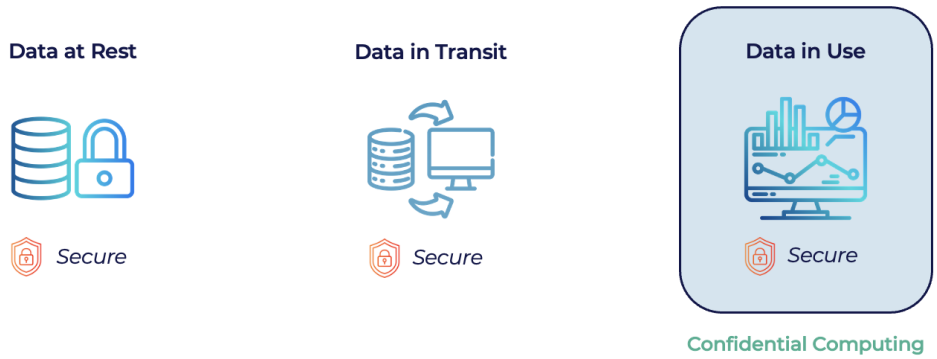

Figure 9: The three different data phases and the implementation of CC

To achieve this, CC uses secure hardware-based isolated environments, called Trusted Execution Environments (TEEs), to process confidential or sensitive data. TEEs create a secure enclave that isolates data processing from the rest of the system, preventing access by other components such as the operating system or hypervisor. This transparency allows AI applications to process plain data within the TEE, avoiding the need to work with encrypted ciphertext, which can be computationally intensive.

Confidential computing initially focused on securing central processing units (CPUs) through technologies like Intel SGX, AMD SEV, and ARM TrustZone. These implementations provided robust protection for serialized tasks executed by CPUs, making them ideal for securing traditional applications. However, AI models, particularly neural networks, require parallel computations due to their reliance on solving large-scale matrix operations. To address this, the scope of CC has expanded to include secure Graphics Processing Unit (GPU)-based computations, leveraging GPUs' superior capability for parallel processing. Figure 10 illustrates a traditional security scope of CC with the expanded isolated environments leveraging GPUs.

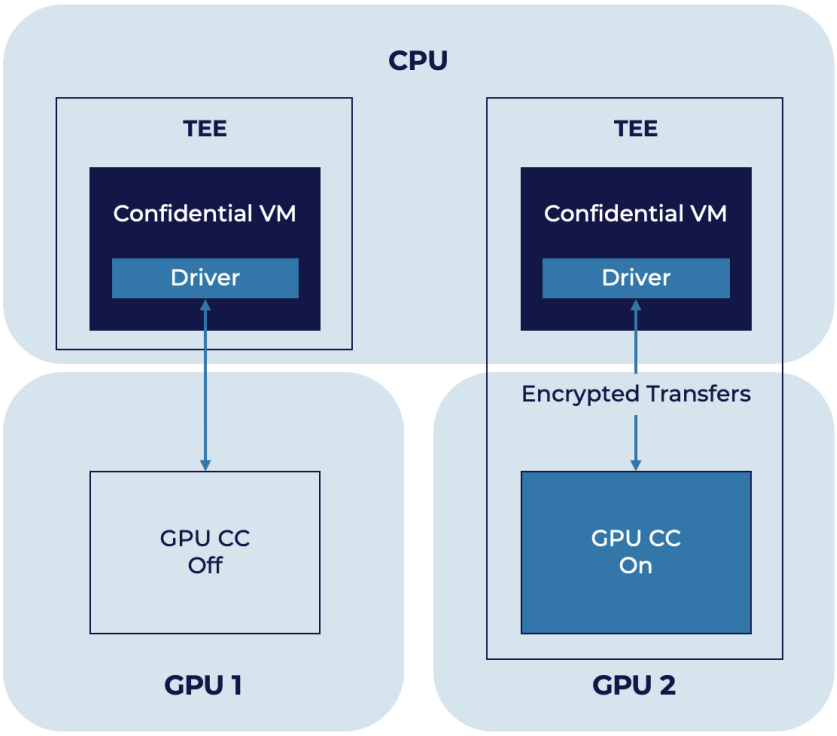

Figure 10: Illustration of a CPU-only (left) and a GPU-extended (right) implementation for confidential computing

Several GPU chip manufacturers have started to explore the field of GPU-based confidential computing, with NVIDIA's Hopper and Blackwell architecture GPU's being one of the first commercially available implementations. They isolate securely GPU-based computations by leveraging the hardware-assisted TEE within their H100 and H200 tensor cores [12].

Confidential computing is considered to date as one of the best solutions for mitigating AI security risks, as it simultaneously protects data in use (training and inference data), secures proprietary model parameters, and protects intellectual property. In addition, CC security is based on hardware capabilities with negligible computational overhead compared to other solutions. This approach ensures that sensitive information remains confidential while providing the high-performance capabilities necessary for modern AI workloads.

Although CC and TEEs are powerful tools to secure AI models, they still have their limitations:

- **Hardware vendor dependency:** Confidential Computing and TEEs rely on hardware vendors (e.g. Intel, AMD, Arm, NVIDIA) to provide security guarantees. As such, if vulnerabilities are found in the hardware or if the vendor's trustworthiness is questioned, it can affect the security model of Confidential Computing.
- **Ad-Hoc development:** The diversity within the TEE landscape results in an ad-hoc development paradigm, meaning that applications designed and developed for a specific TEE implementation are not necessarily compatible with another. This limits flexibility, as it complexifies the adaptation of a different technology.
- **Usage complexity:** TEEs and confidential computing architectures can be complex and difficult to implement, manage, and maintain, particularly for non-experts. However, several solutions do exists that help ease this process, such as CYSEC's ARCA Trusted OS, which provides a scalable and container-friendly CC implementation.

4.6 Remote Attestation

Remote attestation (RA) is a verification process that allows a remote system to prove it is running in a trusted, untampered environment, typically within a TEE [15]. This technology ensures sensitive data and computations, such as AI model training or inference on private data, occur in verified and secure conditions.

The process to achieve RA varies per implementation, but involves the device that requires authentication, called the *attester* or *prover*, to instruct a secure hardware element (e.g. a TEE) to generate signed claims of trustworthiness vouching for the state of the system. These claims are then presented to an external party, called the *verifier*, which appraises whether the claims are originated from a genuine system. After approval by the *verifier*, the attestation results can be forwarded to a *relying party* (i.e. an instance that depends on the correct authentication of the application). An overview of a generic RA workflow can be found in Figure 11.

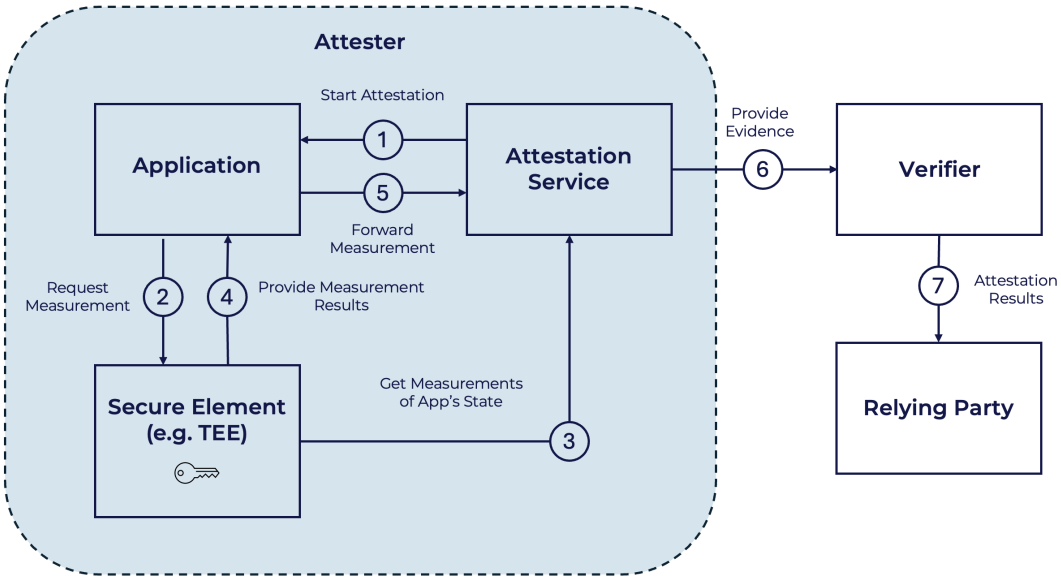

Figure 11: Generic workflow of a RA architecture

In AI deployments, particularly **in cloud or edge environments, remote attestation ensures that the AI model is running in its original, untempered state**. For instance, in financial fraud detection systems deployed on edge devices, RA verifies that the model parameters and logic remain unaltered during deployment, preventing malicious modifications that could compromise the system's accuracy or introduce vulnerabilities.

In Federated Learning, RA ensures that each participant's device is running a genuine and trusted instance of the training code. This prevents adversaries from injecting malicious updates into the aggregated model, preserving the integrity of the training process. For example, RA can validate that only authenticated hospitals contribute to a federated AI model for healthcare diagnostics, ensuring the model remains trustworthy.
Although Remote Attestation is a critical tool for securing AI systems, particularly in distributed and high-risk environments, it does not provide complete security guarantees on its own. Therefore, RA is typically used in conjunction with other security measures, such as Confidential Computing and Secure Multi-Party Computation, to create a comprehensive protection framework. Additionally, RA has several limitations that must be carefully considered:

- **Reliance on Hardware security features:** RA depends on secure hardware elements, such as Trusted Platform Modules (TPMs) or hardware-enforced TEEs. This reliance means that any vulnerability in the underlying hardware can compromise an RA implementation.

- **Complex implementations:** Setting up a robust RA system requires deep expertise in the integration and configuration of the technology stack, including the secure element (e.g., TPM, TEE), and often involves specialized implementations for each device or platform.
- **Vulnerability to replay and man-in-the-middle attacks:** RA itself does not protect against replay and man-in-the-middle attacks, leaving it up to the implementation itself to add proper countermeasures.

4.7 Adversarial Training

Adversarial training is a method used in machine learning to strengthen models against adversarial attacks; malicious manipulations intended to deceive or mislead AI systems. The main concept behind adversarial training is to introduce *adversarial examples* into the training data: specially crafted inputs that are designed to cause a model to make incorrect predictions. By exposing the model to these examples during training, it learns to recognize and handle perturbed data, thereby increasing its robustness and resistance to attacks.

A common approach in adversarial training is to apply techniques such as the Fast Gradient Sign Method (FGSM) [11] or Projected Gradient Descent (PGD) [18]. These methods subtly modify input data in a way that can confuse a model but remain largely undetectable to humans. FGSM, for instance, adds small perturbations to input data in the direction that maximizes the model's loss, making it more likely to cause incorrect outputs. An overview of adversarial training can be found in Figure 12.

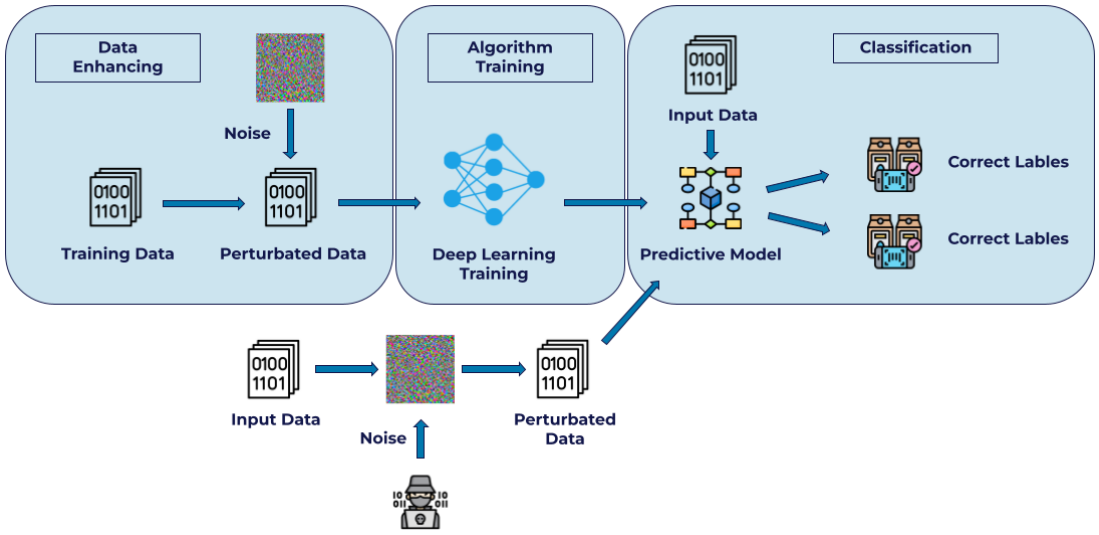

Figure 12: Example implementation of adversarial training of AI models

While the usage of adversarial training for AI models increasing their robustness and resistance to attacks, it does come with some disadvantages:

- **Decreased performance on clean data:** One of the side effects of adversarial training is a reduction in accuracy on clean, unperturbed data. This trade-off between robustness and performance on standard examples occurs because the model's focus shifts to handling adversarial examples, sometimes at the cost of generalization on clean data.
- **Limited robustness generalization:** While adversarial training increases robustness on training data, models often struggle with robustness when facing unseen test data or new attack types.
- **Challenges of high dimensionality:** As data dimensions increase, the distribution of data points becomes sparse, creating blind spots or vulnerabilities where the model's defenses may fail. This makes it especially difficult to apply adversarial training to high-dimensional datasets like those used in computer vision

Table 1 provides a complete overview of the aforementioned security solutions for AI.

Table 1: Overview of AI security solutions

Technology	Strengths and Weaknesses	Available Implementations	Performance Impact	Implementation Type
Homomorphic Encryption Enables computations on encrypted data without decryption	+ High privacy protection - High computational overhead, data size expansion, and reduced model accuracy	Zama Concrete ML, IBM HElayers & HE4Cloud	High computational cost and accuracy reduction	Requires integration within the AI model during operations (training, inference, etc.)
Secure Multi-Party Computation Allows collaborative computation on private data without sharing inputs	+ Strong privacy protection - Limited functionality for complex models, high computational and communication overhead	Meta CrypTen, Duality Secure Data Collaboration Platform	High communication and computation overhead	Transparent to the deployed AI model
Differential Privacy Protects individual data by adding statistical noise to datasets or computations	+ Strong protection of individual data privacy - Reduced data utility and accuracy, challenging parameter tuning	OpenDP, IBM Diffprivlib, Google PipelineDP	Privacy-utility trade-off; accuracy decreases with higher privacy budgets	Requires integration within the AI model
Federated Learning Trains models across decentralized devices without sharing raw data	+ Protects training data privacy - Limited security against gradient leakage, slower model convergence, communication challenges	NVIDIA NVFlare, TensorFlow Federated, Intel OpenFL	Communication-heavy; additional overhead with security measures like DP	Transparent to the deployed AI model
Confidential Computing Protects data during processing using Trusted Execution Environments (TEEs)	+ High security with minimal performance overhead - Hardware dependency and implementation complexity	Intel SGX, AMD SEV, NVIDIA Hopper GPUs, CYSEC ARCA Trusted OS	Minimal performance overhead compared to other methods	Transparent to the deployed AI model
Remote Attestation Verifies AI systems' integrity and secure state during operation.	+ Ensures model authenticity and integrity - Vulnerable to replay attacks; requires hardware integration and complex setup	CYSEC ARCA Remote Unlock, Custom implementations	Dependent on hardware capabilities; potential delays during attestation	Transparent to the deployed AI model
Adversarial Training Exposes models to adversarial examples during training to improve robustness	+ Enhances resistance to adversarial attacks - Reduced accuracy on clean data; struggles with high-dimensional data	Custom implementations using FGSM, PGD	Increased training time; reduced clean data performance	Requires integration within the AI model during training

5 CYSEC Products for AI Security

CYSEC is a cybersecurity company based in Lausanne, Switzerland, specializing in high-security solutions for critical IT infrastructures, with a focus on providing advanced protection for sensitive data and applications, including AI models.

5.1 ARCA Trusted OS

CYSEC's flagship product, ARCA Trusted OS, provides an easy adoption of Confidential Computing, deployable either on premise, or in the cloud. It consists of a hardened Linux-based micro-distribution operating system, offering essential security features, like kernel lockdown, secure boot, an immutable file system and full disk encryption, integrated with hardware-based confidential computing capabilities (e.g., AMD SEV, Intel Arm TrustZone) for enhanced isolation. In addition, it implements a secure containerization platform, which allows users to deploy and manage their containerized applications in a scalable and distributed manner.

This technology offers several key features that can directly benefit AI systems in terms of confidentiality, integrity, and secure execution, particularly in scenarios where models are deployed in distributed environments vulnerable to unauthorized access or tampering.

5.1.1 AI Isolation and Protection from Tampering

At the heart of ARCA Trusted OS lies its robust isolation capabilities, leveraging hardware-backed TEEs to ensure that sensitive AI computations remain tamper-proof. By seamlessly using technologies such as AMD SEV, Intel SGX, ARM TrustZone and extending Confidential Computing capabilities to GPUs like NVIDIA's H100 and H200 with secure TEE support, ARCA Trusted OS creates a secure enclave where critical AI operations, including model training and inference, are executed. This enclave isolates these processes from the rest of the system, including the operating system itself and any other potentially compromised components.

For example, when deploying an AI model for real-time fraud detection in financial transactions, ARCA ensures that the model's parameters, logic, and data are protected against unauthorized access or manipulation. The system also employs secure boot and kernel lockdown to verify the authenticity and integrity of all code at runtime, preventing the execution of malicious modifications. Remote attestation, as explained below, further enhances this capability by allowing external systems to verify that the deployed environment remains in a secure, trusted state. Together, these mechanisms create a robust shield against tampering, ensuring that AI deployments in high-stakes environments operate reliably and securely.

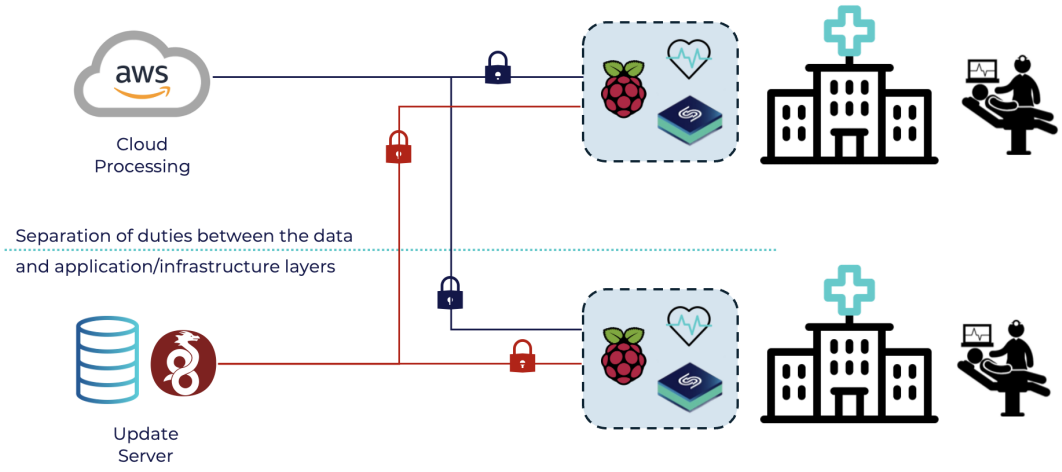

Figure 13: ARCA Trusted OS to secure patient data within Implicity's AI solution

Another example of secure AI deployment has been demonstrated by Implicity, a healthcare company that

leverages distributed architecture to streamline patient data gathering [24]. As shown in figure 13, Implicity uses small devices, based on Raspberry Pi boards running ARCA Trusted OS, to securely collect patient data from apparatus monitoring cardiac information in hospitals. These edge devices serve as trusted end-points that gather sensitive data within a secure environment, ensuring the integrity and confidentiality of patient information at the collection point. The data is then transmitted securely to cloud-based AI services running on AWS for processing, enabling real-time visualization and insights for doctors. The architecture is further supported by a WireGuard server, which manages and updates ARCA Trusted OS and containerized applications, ensuring encrypted communication and seamless updates across the distributed system.

5.1.2 Orchestration of AI Workloads in Distributed Systems

AI workloads are often deployed across distributed environments, from edge devices to centralized cloud systems, requiring sophisticated orchestration to ensure seamless and efficient operations. ARCA Trusted OS incorporates a secure containerization platform that facilitates the deployment, management, and scaling of containerized AI models across distributed clusters. Each container runs in an isolated environment with its dependencies, providing both portability and security for AI applications.

In scenarios such as collaborative emergency management, where AI models at the edge process critical real-time data, ARCA ensures that communications between nodes, whether at the edge or in the cloud, are encrypted, detecting tampering attacks. Secure orchestration allows AI models to synchronize across nodes, ensuring that updates or patches are distributed securely without introducing vulnerabilities. For example, when deploying an updated alert management model across different emergency centers, ARCA enables secure, consistent updates to all edge devices while maintaining operational integrity. A similar use-case described in Figure 14 was setup within an ongoing Horizon Europe Project -COGNIFOG - with our partner Thales France.

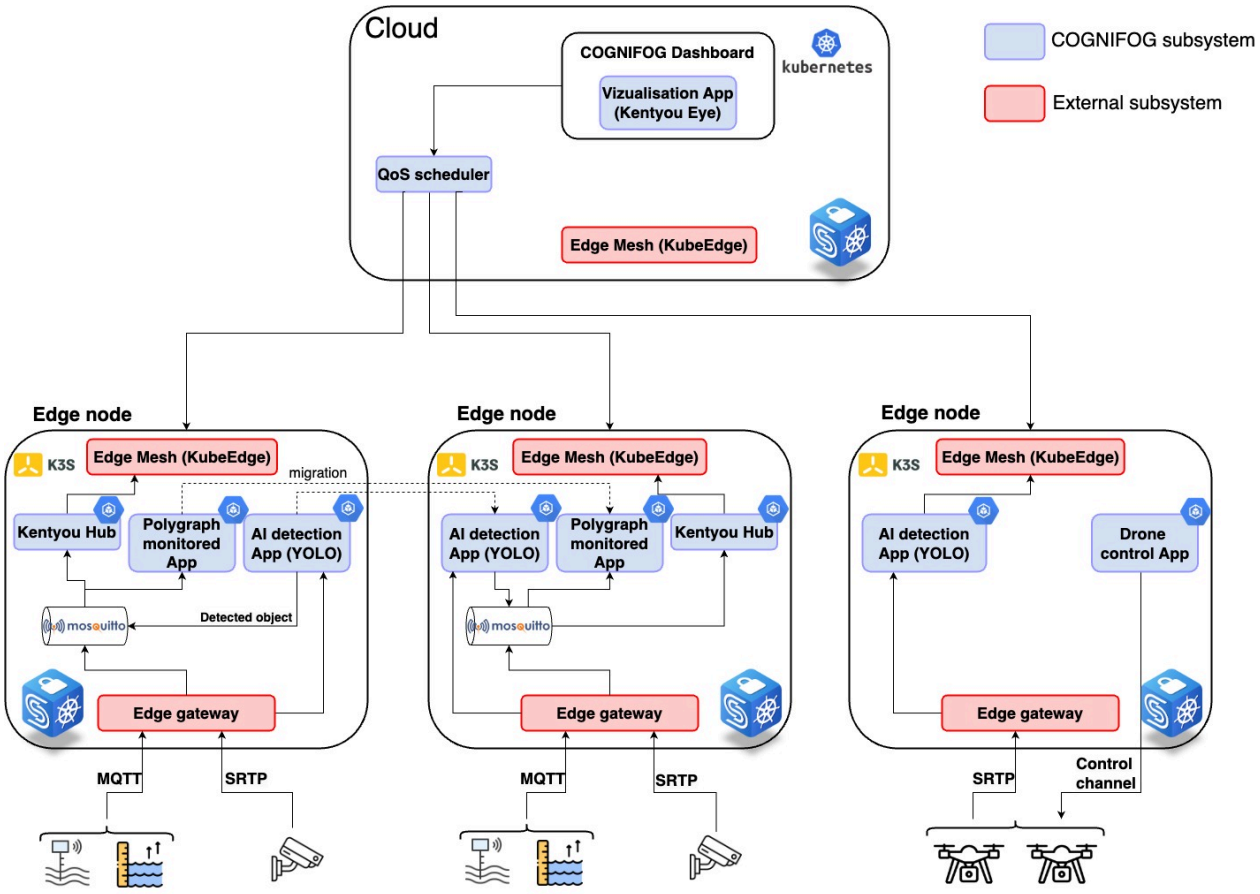

Figure 14: ARCA Trusted OS to secure AI-based Emergency collaborative mission by Thales - COGNIFOG Project

This secure orchestration framework ensures the continuity of AI operations while guaranteeing the resilience of distributed systems, even when spanning diverse environments. By integrating security into every layer of the orchestration process, ARCA Trusted OS mitigates the risks inherent in managing distributed AI workloads.

5.1.3 Scalability of AI Systems

One of ARCA Trusted OS's standout features is its ability to scale AI deployments securely and efficiently. Adding or removing nodes in a distributed system is often a complex and disruptive process, especially when sensitive data or computations are involved. ARCA simplifies this by allowing nodes to be seamlessly integrated or decommissioned without compromising security or disrupting ongoing operations.

For instance, in federated learning networks where participants dynamically join or leave the collaboration, ARCA ensures that new nodes are securely provisioned and authenticated before being added to the cluster. Similarly, when a node is removed, ARCA securely isolates and decommissions it, ensuring that no residual data or credentials remain accessible. This capability is critical in environments such as predictive maintenance in industrial IoT, where new sensors or devices must be frequently added to the network without halting operations.

Moreover, ARCA Trusted OS's integration with Kubernetes enables organizations to dynamically manage their AI workloads across clusters, ensuring that resources are allocated efficiently while maintaining strict security controls. This scalability empowers organizations to adapt to evolving demands, whether expanding operations, responding to increased workload, or scaling down to optimize resource usage, all while ensuring the integrity and security of the overall system.

For more information regarding ARCA Trusted OS, please refer to our website: `https://www.cysec.com/arca-trusted-os/`

5.2 Attested Launch of ARCA Trusted OS

CYSEC aims to enhance the security guarantees of its ARCA Trusted OS solution for AI implementations even further, by integrating a novel and patented protocol called *Attested Launch*. Its objective is to utilize Remote Attestation to ensure that an instance of ARCA Trusted OS can boot if and only if it is authentic and runs in a hardware-based TEE, with both parties being validated by the instance owner. This solution is particularly interesting for cloud-based deployments, as it isolates ARCA Trusted OS virtual machine instances from the cloud host and other tenants. The isolation extends to robust encryption of data across all states — at rest, in transit, and in use.

The attested launch protocol utilizes the full disk encryption feature of ARCA Trusted OS, ensuring that no access to data within the OS is granted until it is decrypted. During boot, the system requests an attestation report from the CC-capable CPU, collecting information about the host's confidential computing setup and the integrity of code executed during the initial phase of VM boot. This report is then sent to a remote verifier, which assesses whether the ARCA Trusted OS instance maintains integrity and operates within a valid hardware-based TEE. Once validated, the remote verifier provides cryptographic keys to decrypt the ARCA OS, enabling it to complete the boot process and establish a secure, trustworthy environment for containerized applications.

The integration of the Attested Launch protocol into ARCA Trusted OS significantly strengthens the security of AI deployments by ensuring that the system can only boot in a validated, secure state. This capability is especially beneficial in several critical AI scenarios as described below.

5.2.1 Cloud-Based AI Model Hosting

In cloud environments, where AI models are hosted and executed, the potential for compromised virtual machines or malicious hypervisors poses significant security risks. With Attested Launch, cloud-based instances of ARCA Trusted OS ensure that only a verified and secure instance can access sensitive hosted AI models

and data. This protects against scenarios where a compromised cloud instance might tamper with the model, leak proprietary parameters, or manipulate inference results.

5.2.2 Distributed AI Workloads Across Multiple Nodes

In distributed AI systems, such as federated learning or decentralized training setups, multiple nodes participate in training or inference tasks. Attested Launch ensures that only trusted nodes are allowed to join the network, protecting the global model from malicious contributions. By verifying the integrity of each participating node during startup, the system prevents attackers from introducing malicious updates or gaining unauthorized access to sensitive data. This is particularly relevant for collaborative AI applications across industries, such as joint healthcare research or financial analytics.

5.2.3 AI Model Deployment at the Edge

Edge devices often process real-time data using AI models in environments that may lack physical security or face potential exposure to tampering. For instance, in autonomous vehicles or industrial IoT devices, Attested Launch ensures that the AI models running on edge devices are deployed securely. By validating the integrity of the OS instance before granting access to model data, Attested Launch prevents adversaries from introducing vulnerabilities into critical decision-making systems.

For more information regarding CYSEC's Attested Launch protocol and a report detailed the exact implementation design, please refer to the following blog post: `https://www.cysec.com/remote-attestation/`

6 Conclusion

Within this white paper, we have highlighted the need for robust cybersecurity implementations for AI, providing examples of the various challenges associated with security.
Data privacy concerns, attacks on the model and/or the data, and IP theft all pose a significant risk for AI systems, requiring advanced security measures. The most prominent of these, including confidential computing, secure multi-party computation, federated learning, remote attestation, and more, have been introduced and discussed as part of this paper, with the aim of providing a detailed yet critical overview of the available security solutions for AI.

However, implementing these solutions requires proactive engagement from stakeholders. Companies must prioritize the security of their AI assets, investing in robust cybersecurity frameworks to protect sensitive data, maintain intellectual property integrity, and build resilience against evolving threats.

If you would like to explore any of the topics discussed in this paper in greater depth or learn how CYSEC can help secure your systems, please contact us at info@cysec.com. Together, we can ensure that your assets remain secure, enabling a future where innovation thrives on a foundation of trust and security.

References

[1] 40+ Important ChatGPT Statistics to Know · Polymer. URL: https://www.polymersearch.com/blog/chatgpt-statistics.

[2] AI Pact | Shaping Europe's digital future. URL: https://digital-strategy.ec.europa.eu/en/policies/ai-pact.

[3] Duality Tech | Secure, Privacy Protected Data Collaboration. URL: https://dualitytech.com/.

[4] PipelineDP. URL: https://pipelinedp.io/.

[5] Zscaler ThreatLabz 2024 AI Security Report | Zscaler. URL: https://info.zscaler.com/resources-industry-reports-threatlabz-ai-security-2024.

[6] IBM/differential-privacy-library, November 2024. original-date: 2019-06-18T13:36:41Z. URL: https://github.com/IBM/differential-privacy-library.

[7] OpenDP, September 2024. URL: https://opendp.org/home.

[8] OpenMined/PySyft, November 2024. original-date: 2017-07-18T20:41:16Z. URL: https://github.com/OpenMined/PySyft.

[9] securefederatedai/openfl, November 2024. original-date: 2021-01-12T21:29:52Z. URL: https://github.com/securefederatedai/openfl.

[10] Ehud Aharoni, Allon Adir, Moran Baruch, Nir Drucker, Gilad Ezov, Ariel Farkash, Lev Greenberg, Ramy Masalha, Guy Moshkowich, Dov Murik, Hayim Shaul, and Omri Soceanu. HeLayers: A Tile Tensors Framework for Large Neural Networks on Encrypted Data. *Privacy Enhancing Technology Symposium (PETs) 2023*, 2023. URL: https://petsymposium.org/popets/2023/popets-2023-0020.php.

[11] Henry Ansah. Adversarial Attacks on Neural Networks: Exploring the Fast Gradient Sign Method, July 2022. URL: https://neptune.ai/blog/adversarial-attacks-on-neural-networks-exploring-the-fast-gradient-sign-method.

[12] Emily Apsey, Phil Rogers, Michael O'Connor, and Rob Nertney. Confidential Computing on NVIDIA H100 GPUs for Secure and Trustworthy AI, August 2023. Publication Title: NVIDIA Technical Blog. URL: https://developer.nvidia.com/blog/confidential-computing-on-h100-gpus-for-secure-and-trustworthy-ai/.

[13] The TensorFlow Federated Authors. TensorFlow Federated, December 2018. original-date: 2018-12-12T23:15:35Z. URL: https://github.com/google-parfait/tensorflow-federated.

[14] Skye Berghel, Philip Bohannon, Damien Desfontaines, Charles Estes, Sam Haney, Luke Hartman, Michael Hay, Ashwin Machanavajjhala, Tom Magerlein, Gerome Miklau, Amritha Pai, William Sexton, and Ruchit Shrestha. Tumult Analytics: a robust, easy-to-use, scalable, and expressive framework for differential privacy, December 2022. arXiv:2212.04133. URL: http://arxiv.org/abs/2212.04133, doi:10.48550/arXiv.2212.04133.

[15] Henk Birkholz, Dave Thaler, Michael Richardson, Ned Smith, and Wei Pan. Remote ATtestation procedureS (RATS) Architecture. Request for Comments RFC 9334, Internet Engineering Task Force, January 2023. Num Pages: 46. URL: https://datatracker.ietf.org/doc/rfc9334, doi:10.17487/RFC9334.

[16] Bennett Cyphers. Understanding differential privacy and why it matters for digital rights, October 2017. URL: https://www.accessnow.org/understanding-differential-privacy-matters-digital-rights/.

[17] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11681878_14.

[18] Simon Geisler, Tom Wollschläger, M. H. I. Abdalla, Johannes Gasteiger, and Stephan Günnemann. Attacking Large Language Models with Projected Gradient Descent, February 2024. arXiv:2402.09154. URL: http://arxiv.org/abs/2402.09154.

[19] Keleno. Secure Multiparty Computing, July 2021. URL: https://medium.com/keleno/secure-multiparty-computing-bd44ee70e1a6.

[20] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. CrypTen: Secure Multi-Party Computation Meets Machine Learning, September 2022. arXiv:2109.00984. URL: http://arxiv.org/abs/2109.00984.

[21] Yunyu Li, Jiantao Zhou, Yuanman Li, and Oscar C. Au. Reducing the ciphertext expansion in image homomorphic encryption via linear interpolation technique. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 800–804, 2015. doi:10.1109/GlobalSIP.2015.7418307.

[22] Omri Soceanu and Ronen Levy. The ultimate tool for data privacy: Fully homomorphic encryption, December 2022. Publication Title: IBM Research. URL: https://research.ibm.com/blog/fhe-cloud-security-hE4cloud.

[23] Holger R. Roth, Yan Cheng, Yuhong Wen, Isaac Yang, Ziyue Xu, Yuan-Ting Hsieh, Kristopher Kersten, Ahmed Harouni, Can Zhao, Kevin Lu, Zhihong Zhang, Wenqi Li, Andriy Myronenko, Dong Yang, Sean Yang, Nicola Rieke, Abood Quraini, Chester Chen, Daguang Xu, Nic Ma, Prerna Dogra, Mona Flores, and Andrew Feng. NVIDIA FLARE: Federated Learning from Simulation to Real-World, March 2023. Publication Title: IEEE Data Eng. Bull., Vol. 46, No. 1 original-date: 2021-07-23T17:26:12Z. URL: https://github.com/NVIDIA/NVFlare, doi:10.48550/arXiv.2210.13291.

[24] Alexandra Vaillant. Securing the ever-evolving healthcare field - A CYSEC & Implicity Partnership, May 2024. URL: https://www.cysec.com/securing-healthcare-field/.

[25] Catherine Wright and Kellin Rumsey. The Strengths, Weaknesses and Promise of Differential Privacy as a Privacy-Protection Framework. *University of New Mexico*. URL: https://math.unm.edu/~knrumsey/pdfs/projects/DifferentialPrivacy.pdf.

[26] Zama. zama-ai/concrete-ml, November 2024. URL: https://github.com/zama-ai/concrete-ml.