Soft Computing Fusion with Applications

www.scfa.reapress.com

Soft. Comput. Fusion. Appl. Vol. 1, No. 4 (2024) 199–219.

Paper Type: Original Article

Bridging AI and Cryptography for Robust Security

Archit Awasthi[1], Pradeep Kumar Singh[1,*], Akhand Pratap Shukla[1]

[1] Computer Science and Engineering (Data Science) (Pranveer Singh Institute of Technology); archit2786@gmail.com; pradeeprudra24@gmail.com; akhandshukla36@gmail.com.

Citation:

Received: 03 April 2024 *Revised: 10 July 2024* *Accepted: 16 September 2024*	Awasthi, A., Kumar Singh, P., & Pratap Shukla, A. (2024). Bridging AI and cryptography for robust security. *Soft computing fusion with applications, 1*(4), 199-219.

Abstract

Cryptography is under mounting pressure from computational requirements, advanced attacks (Notably Artificial Intelligence (AI)-based Side-Channel Analysis (SCA)), and the looming threat of quantum computing to today's standards, such as Rivest–Shamir–Adleman (RSA)/Elliptic Curve Cryptography (ECC). AI has a double-edged potential: It provides potent tools for improving cryptographic design (e.g., S-Box optimization), protocol optimization (e.g., Quantum Key Distribution (QKD)), and security analysis automation, while at the same time empowering intense cryptanalysis. Most importantly, securing AI systems themselves is now a priority [1], quite frequently requiring cryptographic answers like Homomorphic Encryption (HE) and Federated Learning (FL). This paper thoroughly examines the multi-aspect intersection of AI and cryptography. We discuss using various AI methods (Machine Learning (ML), Deep Learning (DL), Evolutionary Algorithms (EAs)) for such constructive and analytical purposes. In addition, we analyze the intensifying threat landscape where AI acts as both threat actors (e.g., advanced malware creation, exploitation of vulnerabilities) and as defenders [2], [3], taking into account the impact of changing regulatory regimes. Emphasizing recent research directions pointing to expansion in areas such as HE and quantum-based cryptography [4], we underscore the vital significance and inherent difficulty of protecting AI systems, solving model resilience, formal verification challenges, explainability requirements (XAI), and implementing secure development methodologies (SecMLOps) [1], [5]. Future research directions need to focus on the creation of AI-resilient cryptographic protocols, quantum-conscious AI security policies, and the general development of reliable AI integration into security-critical applications.

Keywords: Artificial intelligence, Cryptography, Cybersecurity, Post-quantum cryptography, Artificial intelligence security, Cryptanalysis, Privacy-enhancing technologies.

1 | Introduction

In the progressively digitized world we live in, cryptography underpins trust, privacy, and security for communications, transactions, and data at rest. It is in use everywhere; you do not have a choice. However, the traditional cryptographic methods, despite their elementary nature, are under tremendous pressure [6]. The classical computational constraints are problematic in processing large numbers of transactions, and cryptanalysis is in constant development. The rise of usable quantum computing will bring an end to the now-

Corresponding Author: pradeeprudra24@gmail.com

https://doi.org/10.22105/scfa.v1i4.59

ubiquitous asymmetric algorithms, like Rivest–Shamir–Adleman (RSA) and Elliptic Curve Cryptography (ECC), potentially in the short term [7]. Therefore, a complete replacement of cryptographic paradigms is needed.

Into this problematic arena steps Artificial Intelligence (AI), a revolutionary technology fast remaking many fields, among them cybersecurity. Its influence on cryptography is deeply twofold. In one respect, AI provides highly capable tools to bolster and update cryptographic systems [8]. In another respect, AI presents highly effective new means of defeating current cryptographic deployments and brings with it its own important security and privacy risks [9]. As AI technologies become part of mission-critical infrastructure and process huge volumes of sensitive information, protecting the AI models themselves against data poisoning, adversarial attacks, bias amplification, and privacy leakage is not a discretionary add-on anymore but a hard requirement. Notably, cryptographic techniques themselves, such as Homomorphic Encryption (HE) and Federated Learning (FL), are emerging as probable solutions to these AI-based security threats [1].

This complex relationship is exemplified in many important ways, underscoring the multi-faceted function of AI within the cryptographic world:

I. AI and improvement in cryptography: AI, through Machine Learning (ML), Deep Learning (DL), and Evolutionary Algorithms (EAs), is a significant factor in improving cryptography. AI can enhance cryptographic aspects, like designing very complex S-Boxes employed by symmetric ciphers. AI is beneficial in making protocols, such as Quantum Key Distribution (QKD) [7], better by optimizing settings based on the prevailing conditions. AI is also beneficial in identifying new cryptographic methodologies by searching among huge design spaces that are infeasible. Recent progress shows AI's increasing importance in shaping the future of cryptography solutions.

II. AI analyzing and breaking cryptography: In contrast, AI vastly enhances cryptographic breaking. Methods such as Side Channel Analysis (SCA) augmented with DL models to analyze very subtle physical leakages (Power consumption, electromagnetic radiation) can recover secret keys from hardware implementations with high effectiveness, perhaps even evading traditional countermeasures [8]. Rising AI methods could invent even more sophisticated forms of cryptanalysis against both algorithms and implementations.

III. AI for security operations around cryptography: In addition to direct engagement with cryptographic algorithms, AI is increasingly becoming a must-have in security operations dependent on or dealing with cryptographic resources [5]. AI-powered systems enhance threat detection by identifying anomalies in network traffic (Even when encrypted) by analyzing metadata patterns, albeit with limitations, streamline Identity and Access Management (IAM), automate aspects of key lifecycle management, and provide intelligent monitoring of cryptographic infrastructure health [10].

The evolving AI threat landscape: AI is now employed by both good and evil. Evil individuals are utilizing AI to develop malware that can evolve the way it evades detection, and they are conducting highly targeted phishing attacks on a massive scale. They also use AI for automated searching and exploiting weaknesses in systems and misusing advanced models to generate fake data. This fake data can train attack models or make realistic fake videos and images for tricking people [2], [11], [12].

Additionally, we have the issue of quantum computing. Quantum computers might break crucial public-key codes, which means there's a need to establish new Post-Quantum Cryptography (PQC) standards quickly. AI comes in handy here. It can help in designing, testing, and evaluating PQC options, simulate quantum attacks to check their resilience, and even enhance PQC for different platforms [7].

This integration of AI, conventional cryptography, quantum computing, and cybersecurity is fueling a research and innovation boom [4], [5]. Integrating AI is due to immediate security requirements across different domains, demonstrating the need for having and adhering to secure development practices across the lifecycle of AI to mitigate risks [3]. As this technological progress continues, new laws and standards are being created globally. They aim to control AI use, establish security measures, and protect data privacy, which adds more challenges for developers and organizations [11].

With all these complex and ever-changing challenges, this paper aims to take a deep dive into how AI impacts the cryptographic world. It will look at AI's role in creating cryptographic designs, analyzing how they're carried out, and managing security operations. Importantly, the paper will discuss the security risks that come with integrating AI and cryptography. It will cover the threats AI poses to cryptographic systems, the weaknesses inside AI itself, and how cryptographic methods can be applied to protect AI. This detailed overview considers today's threat landscape, current defensive technologies, the ongoing move to PQC, and crucial directions for future research and development to build secure, strong, and trustworthy systems in the era of AI.

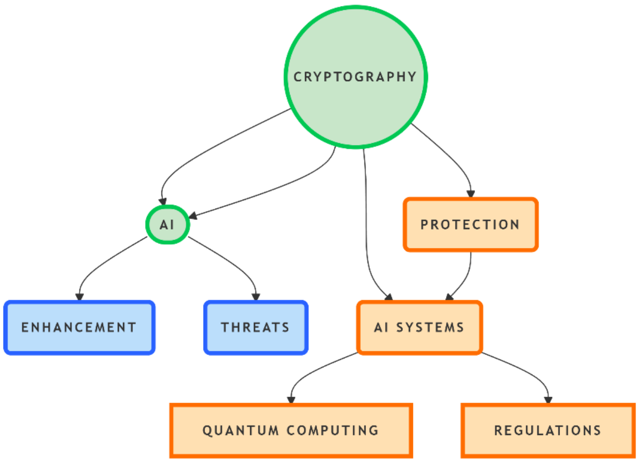

Fig. 1. Conceptual overview of artificial intelligence and cryptography interactions.

2 | Background and Related Work

This section provides the foundations by presenting the basic principles of cryptography and AI that pertain to their intersection. This section also surveys the state-of-the-art as well as recent trends and reveals how these domains interact, reciprocally impact one another, and confront evolving challenges, mainly in terms of security and privacy.

2.1 | Fundamentals of Cryptography

Cryptography provides the essential tools for securing information in the digital age. Key concepts include:

I. Symmetric encryption (For example, Advanced Encryption Standard (AES)): This employs a single shared secret key for both decryption and encryption. The current standard is the AES, which is efficient and speedy, and optimized for encrypting large volumes of data (Bulk encryption). It is a block cipher that operates on fixed-size blocks of data (Normally 128 bits). The central challenge is safely sharing the shared key.

II. Asymmetric encryption (e.g., RSA/ECC): Also referred to as public-key cryptography, this involves the use of a pair of mathematically related keys: A public key (Made publicly available) to encrypt and a private key (Held secretly) to decrypt [5]. Notable examples are RSA, whose security depends on the infeasibility of factoring large prime numbers, and ECC, which relies on the difficulty of the elliptic curve discrete logarithm problem. Although typically slower than symmetric encryption, the asymmetric techniques are essential to conduct secure key exchange (i.e., deriving a symmetric key) and digital signatures, and facilitate authentication and non-repudiation.

III. Hashing (e.g., SHA-256/SHA-3): Cryptographic hash functions return a fixed-size output (Hash or digest) after accepting input of any size. They must be collision-resistant (It is difficult to find two different inputs with the same output) and one-way (Reversing them computationally is not possible). Hashes are core to ensuring data integrity (Identification of modifications) and storing passwords securely (Keeping hashes rather than plain text). SHA-2 and SHA-3 are now popularly utilized families of hash functions.

IV. PQC concepts: The future arrival of very powerful quantum computers jeopardizes the security hypotheses of today's widely deployed asymmetric algorithms, such as RSA and ECC (Because of Shor's algorithm). PQC seeks to create new cryptographic primitives, specifically for public-key operations (Key exchange, signatures), that will be secure against attacks by both classical and quantum computers. Large categories being developed by NIST include lattice-based, code-based, hash-based, and multivariate cryptography.

Key cryptographic techniques for AI Security: As AI models process vast and often sensitive data, specific cryptographic techniques are crucial for enhancing their privacy and security [1]:

I. HE: Enables direct computation on encrypted data without decryption beforehand. It is very pertinent to privacy-preserving AI since it allows third parties (Such as cloud providers) to train ML models or do inference on sensitive data without ever accessing plaintext data. The biggest drawback now is high performance overhead.

II. FL: A distributed ML that has models trained locally on devices that possess the data (e.g., user phones, enterprise servers). Rather than uploading raw data to a central server, model updates (e.g., gradients) are sent. While mostly an architectural design to maintain privacy, FL typically also incorporates cryptographic techniques like secure aggregation (Using mechanisms like Secure Multi-Party Computation (SMPC) or Differential Privacy (DP)) to protect the individual model updates at aggregation.

III. Zero-Knowledge Proofs (ZKPs): Allow a party (The prover) to convince another party (The verifier) of the truth of a statement, without revealing anything except the fact that the statement is true. Uses of AI that could apply include demonstrating correct model inference without disclosure of the model's internal parameters or the input data, or checking properties of a dataset without disclosure.

IV. DP: A formal system for yielding quantifiable privacy assurances, typically by introducing calibrated noise to query results on a database or to ML model outputs. Even though it is not a cryptographic technique in its own right, DP is often used in conjunction with cryptographic methods (e.g., secure aggregation in FL or computation in HE) to ensure that the final output of an AI system does not reveal sensitive information about individual data points used in its training or analysis. It augments crypto by offering output privacy assurances.

V. Quantum Cryptography (QKD): Unlike PQC, QKD uses principles of quantum mechanics (Such as the Heisenberg uncertainty principle and the no-cloning theorem) to enable two parties to share a common secret key. Protocols such as BB84 seek to make it such that any measurement by an eavesdropper on the quantum signals that transmit the key information will necessarily cause detectable interference. Its security is based on physical laws and not computational hardness. Real-world challenges involve limits of transmission distance and incorporation in current infrastructure [7].

Table 1. Key cryptographic techniques for artificial intelligence security.

Technique	Core Principle	AI Security Use Case	Key Limitation
HE	Compute on encrypted data	Privacy-preserving ML training/inference	Performance overhead
FL	Train locally, share updates only	Decentralized training, data minimization	Needs crypto for updates
ZKPs	Prove the statement's truth, reveal nothing else	Secure model inference proof; data checks	Complexity; overhead
DP	Quantifiable privacy via noise	Limit leakage on individual training data	Privacy vs. utility trade-off
QKD	Physics-based secure key exchange	Secure comms channel keys for AI data	Distance limits; hardware

2.2 | Fundamentals of Artificial Intelligence

AI involves a variety of methods that allow machines to execute tasks usually involving human intelligence. Subfields of relevance include:

I. ML: Programs that learn data patterns without the programming of each rule.

II. Supervised learning: Labelled data is trained on (Input-output pairs). Most routine tasks are classification (Allocation of inputs into previously defined categories, e.g., spam classification) and Regression (Forecasting continuous numerical outputs, e.g., forecasting network delay). Support Vector Machines (SVMs), Decision Trees, and Neural Networks (NNs) are some of the most popular algorithms.

III. Unsupervised learning: It learns from unsupervised data. A primary task is clustering (Similar data points grouped, e.g., discovering communities in network traffic).

IV. DL: A branch of ML that predominantly employs artificial NNs, having more than a single layer as deep architectures. The main architectures are the following:

- *Multi-Layer Perceptrons (MLPs): Simple feedforward networks.*
- *Convolutional Neural Networks (CNNs): Strongly capable in the case of grid-shaped data (For instance, images) and detecting space hierarchies of features. They're utilized on sequence data (Such as side-channel power traces or network traffic) by approaching sequences as a 1D grid to learn about local structures [8].*
- *RNNs and LSTMs: Designed for sequential data in particular, e.g., text or time series, these networks possess an internal state that operates on sequences. The sequential data can prove helpful when attempting to identify patterns in network traffic or where crypto leakage is time-dependent.*
- *Generative models: They attempt to create fresh samples of data that are like a provided set of data. One extremely popular example is Generative Adversarial Networks (GANs), where a generator network produces samples and a discriminator network attempts to distinguish real and fake samples. These can be used for data augmentation but pose risks in creating realistic counterfeit attacks or data [12].*
- *EAs: Optimization algorithms inspired by biological evolution. Methods, such as Genetic Algorithms (GAs), use populations of candidate solutions, where they are subjected to selection, crossover, and mutation over generations, to discover good-quality solutions to complex optimization problems, such as optimizing cryptographic components, like S-boxes, in, for instance, [7].*
- *Reinforcement Learning (RL): An agent learns to produce a sequence of decisions through acting in an environment and receiving rewards or penalties. Applied to optimization in dynamic environments, e.g., possibly optimizing parameters for QKD protocols in response to real-time channel feedback [7].*

2.3 | State-of-the-Art and Recent Trends

The intersection of AI and cryptography is a dynamic field marked by rapid advancements, driven by both the potential of AI to enhance cryptographic methods and the emergence of AI as a powerful tool for both attackers and defenders. This section synthesizes the current landscape across key application areas.

2.3.1 | Artificial intelligence in cryptanalysis: Sharpening the attacker's toolkit

In the past, cryptanalysis depended on mathematical intuition and computational brute force. AI, and particularly ML and DL, brings strong pattern recognition capabilities that far exceed traditional methods and are more likely to concentrate on implementation weaknesses rather than mathematical attacks.

I. AI-augmented SCA: This is probably the most sophisticated application of AI in crypto implementation attacks. Profiled SCA, in which an attacker profiles a controlled "profiling" device identical to the victim's, heavily depends on AI [8].

- *Mechanism: Large datasets of leakage traces (Power consumption, electromagnetic emanations) are annotated with the intermediate results of the target cryptographic operation or key bytes and are employed to train DL models (Most commonly CNNs because of their ability to process raw trace data and identify spatial/temporal patterns, or MLPs that are trained on pre-processed features).*

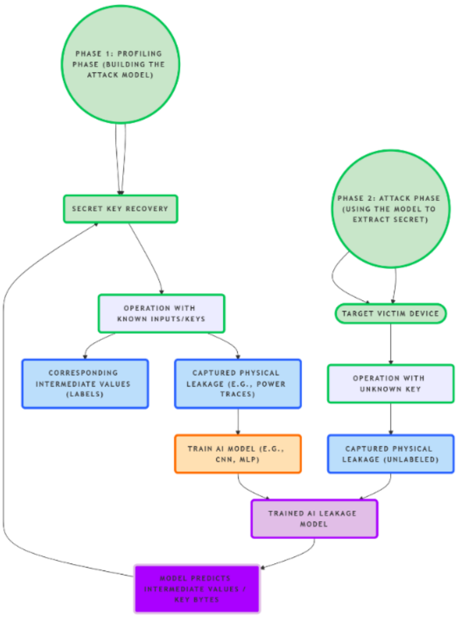
PHASE 1: PROFILING PHASE (BUILDING THE ATTACK MODEL)
SECRET KEY RECOVERY
PHASE 2: ATTACK PHASE (USING THE MODEL TO EXTRACT SECRET)
OPERATION WITH KNOWN INPUTS/KEYS
TARGET VICTIM DEVICE
CORRESPONDING INTERMEDIATE VALUES (LABELS)
CAPTURED PHYSICAL LEAKAGE (E.G., POWER TRACES)
OPERATION WITH UNKNOWN KEY
TRAIN AI MODEL (E.G., CNN, MLP)
CAPTURED PHYSICAL LEAKAGE (UNLABELED)
TRAINED AI LEAKAGE MODEL
MODEL PREDICTS INTERMEDIATE VALUES / KEY BYTES

Fig. 2. Flowchart of artificial intelligence-enhanced profiled side-channel attack.

- *Effectiveness: These models tend to surpass the conventional statistical attacks (Such as Differential Power Analysis (DPA), or Correlation Power Analysis (CPA)), needing much fewer attack traces to extract the secret key. They show greater robustness against usual countermeasures such as masking and noise injection, because DL models can learn to remove noise or detect leakage even when distributed across multiple shares (In masking schemes).*
- *Challenges: The major challenges are still present, such as the necessity for accurate trace synchronization (Though some AI methods are trying to counteract this), a lot of data necessary for good profiling, and the "portability" issue – a model that has been trained on one physical chip does not necessarily run well on an equivalent chip due to slight differences in manufacturing (Process variation). Work on domain adaptation and transfer learning tries to counteract this [8].*

II. AI exploitation by adversaries (Beyond SCA): The Greater availability and advancement of AI tools pose wider security threats to cryptographic-reliant security systems [2].

III. Automated vulnerability discovery: AI methods (Such as RL or generative models in combination with fuzzing) are being researched for automatically finding implementation bugs or logical errors in cryptographic libraries and protocols.

IV. Optimized attack plans: AI could optimize parameters for established attacks (e.g., brute-force variants, differential attacks) or even directly search for completely novel cryptanalytic vectors by searching for non-randomness or attackable biases in cryptographic outputs.

V. Increased social engineering and evasion: Generative AI [12] can create extremely realistic deepfake audio/video for social engineering attacks to hijack credentials or keys. It can be used to create polymorphic malware that modifies its signature dynamically to avoid detection by conventional (And even many AI-driven) security solutions defending systems where cryptography is being utilized. AI can also create network traffic patterns specifically tailored to bypass AI-driven intrusion detection systems that are monitoring encrypted communications.

2.3.2 | Artificial intelligence in crypto design and optimization: Building better defenses

Since AI becomes an intrinsic part of various systems, even possibly security-related ones, ensuring the AI models themselves are safe from targeted threats is crucial. If AI models are not secure, they can jeopardize the very security that they are supposed to enable [1].

Key vulnerabilities:

I. Data poisoning: Attackers covertly tamper with the training data to insert backdoors or biases, which makes the AI model intentionally fail on certain inputs in the future (e.g., always passing a particular kind of malware).

II. Evasion attacks: Adversaries design malicious inputs at inference time that are slightly perturbed from valid inputs but make the AI model misclassify (e.g., making malware appear harmless to an AI detector).

III. Model extraction: Attackers attempt to steal intellectual property or facilitate future attacks by attempting to replicate the parameters or architecture of a deployed model earlier (Usually via an API).

IV. Privacy attacks (Membership inference): In this instance, privacy attacks refer to efforts to determine whether the training data for the model contained information from a specific user.

V. Cryptography and other mitigation techniques: To maintain data confidentiality against the AI model operating entity, HE and SMPC enable training or inference on encrypted data [1].

VI. FL: Reduces exposure to raw data through local training. May necessitate cryptographic secure aggregation methods to secure individual model updates when aggregated centrally.

VII. ZKPs: Potentially used to demonstrate properties about a model or its output (e.g., "this output was produced according to the agreed protocol") without exposing sensitive internals.

VIII. Non-crypto defenses: Techniques such as adversarial training, input validation, model watermarking, and DP (For output privacy) are essential components of a layered defense.

2.3.3 | Secure Artificial intelligence development practices (MLSecOps/AISecOps)

To spot the flaws in AI, we need to make security a key part of its development and use, not something added later.

From start to finish: Just like with DevSecOps in software, security should be woven into every step. Security means gathering and processing data securely to avoid bias and protect privacy, training models with a focus on security by testing them against tricky examples, checking their performance, security, and fairness thoroughly, deploying securely by reinforcing APIs and controlling access, and keeping an eye on things once they’re in action to pick up any issues.

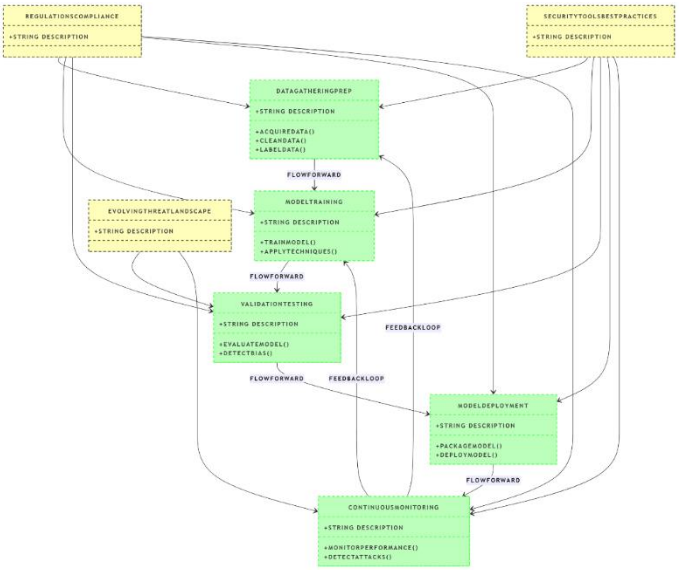

Fig. 3. Integrating security throughout the machine learning lifecycle.

Why it matters: Following these steps is essential for building AI systems that people can trust. Trust is especially true for AI used in critical security situations where errors can have serious effects.

2.3.4 | Regulatory landscape: Shaping artificial intelligence and crypto deployment

The legal and ethical landscape surrounding AI and data protection is rapidly evolving, directly influencing the employment of cryptographic techniques and AI.

Data protection legislation: Regimes like GDPR (Europe) and the majority of US state law (e.g., CCPA) impose severe conditions on handling personal information, influencing AI system design and driving the use of privacy technology like Privacy-Enhancing Technologies (PETs), including cryptography like HE and FL [11].

AI-specific rules: Models such as the EU AI Act classify AI systems by risk level, subjecting high-risk applications to stricter requirements (Such as security, transparency, and human intervention) [11]. The NIST AI Risk Management Framework offers guidance to organizations for responsibly managing AI risks [11].

Impact: Compliance requires diligent attention to data handling, model security, explainability, and cryptographic techniques employed to secure data both at rest, in transit, and computation (Where HE comes into play). Mastering this intricate web of regulations is a considerable hurdle for organizations to overcome.

3 | Methodology: Applications of Artificial Intelligence Techniques in Cryptography

AI offers a sophisticated toolkit that transcends traditional methods, enabling novel approaches to understanding, building, and managing cryptographic systems. This section elaborates on the specific AI methodologies employed, dissecting how they function within the context of cryptanalysis, design, and operational security.

3.1 | Background and Related Work

AI's ability to learn complex correlations and identify subtle patterns from vast datasets makes it particularly effective in areas where classical cryptanalysis struggles, such as implementation-level vulnerabilities or statistical leakage.

3.1.1 | Background and related work

The related work is one of the most significant and well-documented uses of AI in cryptanalysis. The technique is aimed at learning the correspondence between physical emanations and confidential internal computations.

In-depth exploration of modelling:

I. Why CNNs: Power and EM traces are time-series signals where the relevant leakage tends to appear as particular shapes or patterns at particular (Though possibly slightly shifted) time points. CNNs, with their convolutional filters, are particularly good at learning spatially (In this case, temporally) local patterns regardless of their precise position. 1D CNNs are typically employed, learning filters that recognize characteristic peaks, dips, or shapes corresponding to particular operations (Such as an S-Box lookup or Hamming Weight (HW) processing).

II. Why MLPs: MLPs can work well too, particularly when traces are appropriately aligned, or if there has been pre-specified feature selection (Points of Interest (PoIs)). They learn intricate non-linear mixtures of input points (Voltage/EM readings at a given time).

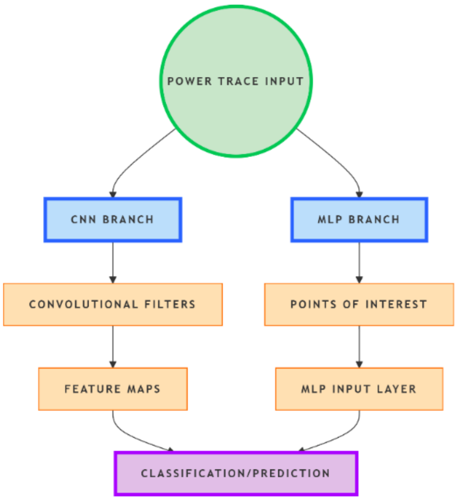

Fig. 4. Conceptual flow of power trace analysis using a convolutional neural network and multi-layer perceptron branches.

III. Feature engineering vs. End-to-end: Previous methods may have included manual feature engineering (Picking certain PoIs manually using domain understanding) before providing data to less complex ML algorithms. DL, especially CNNs, tends to work end-to-end using raw or lightly pre-processed traces as input and allowing the network to find the useful features on its own.

IV. Profiling data and labelling nuances: The intermediate value targeted (The label) is often the HW or Hamming Distance (HD) of an internal state, as this often correlates better with power consumption than the actual value itself. The model might predict one of 9 possible HW values (For an 8-bit state) or directly predict the full 256-class intermediate state value, often using a SoftMax output layer.

V. Addressing misalignment: Techniques beyond robust model architectures (Like CNNs) include:

- *Trace pre-processing: Attempting to align traces before training or attack by correlation-based alignment techniques or Dynamic Time Warping (DTW).*
- *Attention mechanisms: Incorporating attention layers into DL models allows the network to selectively concentrate on the most informative time steps in a trace, hence implicitly compensating for some misalignment.*
- *Model transferability strategies: Typically, reducing device variability involves:*

- *Domain adaptation methods: They encompass adversarial domain adaptation, where the model is trained on features that are invariant to devices but predictive of the label.*
- *Transfer learning: The model is fine-tuned first on a highly limited labelled corpus from the specific victim device, following pre-training on a large corpus from one or more profiling devices.*

VI. Beyond profiling: Other than profiling being the most typical application of AI in SCA, research is also looking into AI for non-profiled SCA. Profiling is more challenging since the labels are not known. Methods could include unsupervised learning (Grouping traces to discover exploitable differences), semi-supervised learning (Assuming partial information exists), or merging AI feature extraction with traditional DPA/CPA correlational steps.

3.1.2 | Automated vulnerability finding (Implementation level): Identifying bugs and flaws

Here, AI assists in rigorously testing cryptographic software/hardware implementations, moving beyond just functionality checks to uncover security vulnerabilities.

I. Methodology - intelligent fuzzing:

- *AI-powered coverage-guided fuzzing: Utilize ML models to determine which of the mutated inputs should probably be hitting "interesting" functions involved in error-handling or cryptographic computation, or maximizing code coverage (Taking new paths).*
- *GAN-based seed generation: GANs can be trained on pre-existing inputs (Such as valid protocol messages or test vectors) and trained to produce new inputs that may be malformed but are syntactically valid. GANs can cause parsers or state machines in crypto libraries to behave improperly.*
- *ML crash triage: Sort crashes or hangs that are discovered through fuzzing, grouping related root causes, or giving priority to those that are more likely to affect security (e.g., memory corruption vs. simple null dereferences).*

II. Methodology - augmenting formal methods:

- *Learning invariants: ML can potentially automatically learn probable program invariants—conditions that never change—by examining execution traces. If such inferred invariants were violated during testing, it would be a sign of bugs.*
- *Using ML models trained on code features (e.g., graph NNs on control flow graphs), path prioritization in symbolic execution predicts which paths are likely to have vulnerabilities (Based on complexity heuristics or historical data). ML models help the symbolic execution engine to be directed more effectively than random or depth-first search.*
- *Implementation-specific side channel detection: Although more difficult than simple fuzzing, theoretically, one could train models to detect code patterns (e.g., data-dependent conditional branches, variable-time operations) that are known to be predictive of potential timing or cache side-channel vulnerabilities.*

3.2 | Artificial Intelligence-Assisted Cryptographic Design and Optimization: Crafting and Refining Security

Leveraging AI's search and optimization capabilities to create novel cryptographic elements or fine-tune existing protocols.

3.2.1 | Optimizing cryptographic primitives: Searching the design space

Finding optimal S-Boxes, permutation layers, or other components with specific resilience properties.

EAs in depth:

I. Representation: How would an S-Box (Say, an 8x8 look-up table) be represented as a chromosome? It might be an explicit list of output values or the parameters of a generating function.

II. Fitness function information: Should be created with caution to achieve a balance among several conflicting cryptographic characteristics. The weights (w1.w4) are extremely significant.

III. Search algorithms: Aside from simple GAs, other algorithms like Particle Swarm Optimization (PSO) or simulated annealing could also be used to scan the component design space.

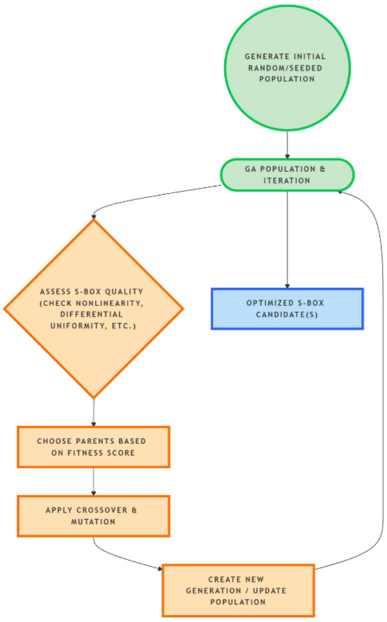
GENERATE INITIAL RANDOM/SEEDED POPULATION
GA POPULATION & ITERATION
ASSESS S-BOX QUALITY (CHECK NONLINEARITY, DIFFERENTIAL UNIFORMITY, ETC.)
OPTIMIZED S-BOX CANDIDATE(S)
CHOOSE PARENTS BASED ON FITNESS SCORE
APPLY CROSSOVER & MUTATION
CREATE NEW GENERATION / UPDATE POPULATION

Fig. 5. Genetic algorithm process for S-Box optimization.

IV. Requirement for validation: Strong validation requirement for significant post-hoc cryptanalysis (Algebraic, differential, linear attacks) of AI-hacked contenders, as the AI will have only optimized in terms of the metrics to the fitness function and could have neglected other areas of attack.

V. Learning-based methods: NNs might learn to estimate computationally costly property evaluations (Such as evaluating non-linearity), accelerating the fitness function evaluation within an EA environment. Generative models (Such as autoencoders or GANs) may be learned on pre-existing known-good primitives to pick up underlying patterns and subsequently create new candidates possessing those desired patterns.

3.2.2 | Protocol parameter tuning: Adapting to the environment

Dynamically optimizing protocol performance based on real-time conditions.

I. QKD RL: Channel statistics (Error rate, signal-to-noise ratio, estimated presence of eavesdropper) may be used to construct the "state" of an RL agent. The parameters that can be changed are called "actions" (e.g., setting privacy amplification strength, choosing error correction block size). The computed secure key rate obtained in a period is the "reward." To maximize the long-term cumulative reward (Total generated secure key), the agent learns a policy (State -> Action).

II. ML prediction for configuration: Supervised ML models, such as regression models, are trained from simulation or experimental data to predict the best parameter set given input channel metrics. The best configurations are determined offline.

III. Security trade-offs: When making dynamic adjustments to parameters according to channel conditions, security implications need to be considered with caution. For instance, reducing error correction to enhance key rate will render the system susceptible if an attacker exploits the channel error characteristics as well. The tuning policy of the AI should also be resilient.

3.2.3 | Artificial intelligence in post-quantum cryptography candidate evaluation

Assisting with the complex process of selecting future-proof PQC standards.

I. Predictive modelling: Features extracted from the mathematical structure of PQC candidates, such as lattice basis properties, code parameters, and polynomial characteristics, are used to train ML models that estimate computational hardness or forecast a candidate's susceptibility to known attacks. Setting priorities for human analysis could be facilitated by this.

II. By using AI-driven automation to run known attack implementations (Such as information set decoding and lattice reduction algorithms) against a variety of PQC candidate parameters, Attack Simulation at Scale can potentially identify edge cases or parameter selections where security fails more quickly than anticipated.

3.3 | Artificial Intelligence for Enhancing Security Around Crypto Implementations: Operational Intelligence

Applying AI to manage the use of cryptography and monitor the context in which it operates, rather than altering the crypto itself.

3.3.1 | Intelligent Key Management: Context-Aware Security

Transcending fixed key rotation schedules to dynamic, risk-based management.

Details of the methodology: Uses techniques like threat intelligence-based predictive analytics (e.g., raising risk for keys associated with protocols that recent attacks have targeted), anomaly detection on key usage logs (Detecting outliers from learned user/system profiles), and potentially clustering algorithms to categorize keys by usage context or risk level. The output may be recommendations entered into Key Management Systems (KMS), automated revocation workflows, or alerts. Requires reliable input from IAM data.

3.3.2 | Anomaly detection in encrypted traffic: Reading between the encrypted lines

Deriving security information from patterns and metadata without the requirement for decryption keys.

Feature space: Temporal aspects (Packet timings, bursts, periodicity) and volumetric aspects (Bytes sent up/down, flow duration) are primary features, complemented by protocol artifacts (Selected cipher suites, TLS versions, certificate information such as issuer, age, self-signed, JA3/JA3S hashes) and network context (Pre-flow DNS queries, IP reputation, geolocation).

Table 2. Features for security analysis of encrypted network traffic.

Category	Examples	Potential Security Insight
Temporal	Timing patterns, duration, frequency	Detects C&C beaconing, timing anomalies
Volumetric	Data volume, packet sizes	Reveals data exfiltration, unusual sizes
Protocol	TLS details, Cipher Suite, Certs, JA3	Finds weak crypto, fingerprints clients/malware
Network	IP/DNS context, IP reputation, GeoIP	Links to malicious infrastructure, anomalies

Modelling techniques:

I. Supervised: Requires labelled sets of malicious encrypted traffic (e.g., C2 from known malware families). Train classifiers (e.g., Random Forests, Gradient Boosting, LSTMs, CNNs) on extracted features. Good for known threats, but is vulnerable to novelty.

II. Unsupervised: Methods such as autoencoders are learnt to reconstruct "normal" traffic feature vectors; a large reconstruction error indicates an anomaly. Clustering (K-Means, DBSCAN) may cluster similar traffic flows, where outliers might be malicious. Statistical approaches (Modelled feature distributions) are also used. More efficient at detecting novel threats but tend to produce more false positives.

III. Deployment context: Usually deployed on network sensors (e.g., Zeek/Bro, Suricata) that pull the appropriate metadata without decrypting the payload. Success heavily relies on the particular threat one is attempting to detect (e.g., detection of TOR may be simpler than detection of nuanced C2 channels intended to be inconspicuous).

4 | Security Analysis and Challenges

The intersection of cryptography and AI requires a thorough and broadened security analysis, though promising. The PQC-NN case, based on McEliece, is a good case study. We need to examine its resistance to conventional cryptographic attacks, critically evaluate the novel vulnerabilities introduced by its AI aspect, comprehend the growing threats from adversarial AI, and tackle the major practical issues in its development, deployment, and regulation.

4.1 | Cryptographic Resistance: Foundations and Artificial Intelligence Impact

I. Classical security foundations: The baseline security claim of the PQC-NN relies squarely on the assumed intractability of the syndrome decoding problem for general linear codes, on which the McEliece cryptosystem is founded. This problem is NP-hard, in that there is no known efficient classical algorithm to solve it in the general case for well-chosen parameters. Attacks usually aim at either recovering the private key structure (Structural attacks) or directly decoding ciphertexts (Message-space attacks). The first NN mapping mainly converts this structure; thus, the underlying difficulty is still there.

II. Impact of non-linearity: Introducing non-linear activation functions complicates the picture. While these functions don't fundamentally weaken the hardness of the underlying syndrome decoding problem against an attacker who knows the exact, intended non-linear operation, they significantly obscure the linear algebraic structures inherent in basic McEliece. Non-linearity can thwart a specific class of classical cryptanalysis that relies explicitly on exploiting linear relationships between plaintext, ciphertext, and key elements. But these non-linearities make formal analysis harder as well and could, in unintended ways, impart subtle statistical bias or other attack vectors if they are not carefully selected and deployed.

III. Quantum resistance: The fundamental strength of a McEliece-based system is its ability to resist attacks from quantum computers. Unlike RSA and ECC, whose security foundations (Factoring and discrete logarithms) are broken by Shor's algorithm, there are currently no known efficient quantum algorithms that provide a significant speedup for solving the Syndrome Decoding problem for the types of codes typically used in McEliece. Quantum resistance makes the PQC-NN concept inherently post-quantum secure at its cryptographic foundation, addressing a major future threat.

4.2 | Artificial Intelligence-Related Vulnerabilities and Threats: A New Attack Surface

The integration of AI transforms the cryptographic primitive into an ML model, opening it up to a distinct set of potent threats.

Vulnerabilities against the crypto process (Via AI-enhanced attacks):

I. AI-enriched SCA: Information leaks from any physical realization. Isolating informative signals from noisy side-channel traces is a strength of AI, and DL in particular [8]. Parts of the network parameters (Weights) or intermediate results may be revealed by power variability or electromagnetic radiation from matrix multiplications or activation function computations inside the NN. Although the PQC-NN's noise perturbation (r) and non-linearities might cause some unintentional obfuscation, they won't take the place of dedicated, moral SCA countermeasures (Like masking or specific hardware logic styles). Before deployment, rigorous, cutting-edge AI-based SCA must be used for empirical testing to see if these embedded features offer any detectable resistance or if traditional countermeasures are still required. The risk is further increased by the demonstrated ability to transfer SCA attack models between devices [8].

Direct vulnerabilities of the AI model itself: The PQC-NN, as an AI artifact, is vulnerable:

I. Model extraction: An attacker could use query access (Even black-box) to train a "substitute" model that approximates the functionality of the PQC-NN. If this replacement model converges close enough, examination of its parameters may elicit the underlying crypto structure (S, G, P^1, R, S^1 equivalent weights), resulting in full system compromise. The challenge is the amount of query access required and how well the heart of the crypto can be approximated.

II. Evasive inputs: Strategically designed inputs (Malformed ciphertexts upon decryption) might take advantage of the NN's non-linearities or learned bounds to bring about specific misclassifications and thereby result in false decryptions or even skip past error-correction methods in novel ways. This attack avenue tests the discrepancy between the mathematically perfect cryptographic function and the NN's learned approximation.

III. Data poisoning and backdoors: Affecting the training data, though perhaps more difficult for the central cryptographic mapping that must be deterministic, is still a threat, particularly if there is transfer learning or FL applied to optimize or deploy. An attacker may secretly poison training data to embed backdoors – particular inputs that will cause deterministic failure or leak essential information, or make the general robustness of the model suffer [1].

IV. Membership inference: The attackers can try to determine if a particular bit of information was used in the NN's training set, representing a privacy threat if the training was performed on sensitive cryptographic data or usage patterns.

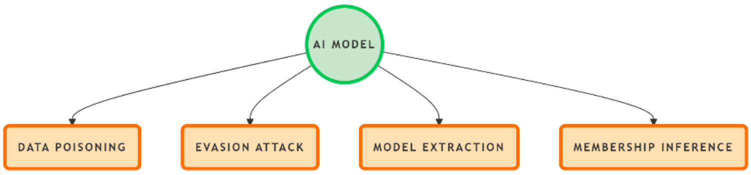

Fig. 6. Common attacks against artificial intelligence models.

Emerging threats FROM malicious AI: The security landscape is further complicated by attackers using AI:

I. Deep AI cryptanalysis: Beyond SCA, future advanced AI models can recognize slight statistical weaknesses or unfairness from the specific NN topology, activation functions, or training procedure, leading to new analytical attacks that were not predicted by traditional methods.

II. Using generative models to detect vulnerabilities: Deep generative models like GANs can produce extremely realistic synthetic data that's applicable in cryptography. They help detect unusual situations or verify how PQC-NNs behave under errors and what kind of response they yield [12]. They can also create realistic side-channel traces, which help make better Side-Channel Attack models.

III. AI for automatic vulnerability detection and exploit creation: AI platforms are powerful tools for automating the detection of vulnerabilities and code scanning. With AI, it's possible to quickly review PQC-NN cases in both software and hardware description languages to find timing issues, hidden bugs, or resource usage vulnerabilities far faster than manual analysis could achieve [2].

4.3 | Challenges: Bridging Theory, Practice, and Governance

Realizing the potential of AI-integrated cryptography requires overcoming substantial hurdles:

I. Computational and training complexity: PQC schemes, such as McEliece, have traditionally depended upon extremely large matrices (Parameters). McEliece matches millions or billions of NN weights that take a lot of time to train, as well as enormous computational resources (GPU/TPU clusters). Furthermore, large, carefully constructed datasets specific to the cryptographic problem and possible attack paths might be needed for NN adjustment or training the supporting AI elements, although fresh data is not needed when extrapolating the established algorithm.

II. Lack of formal security proofs: The foundation of cryptographic trust is formal mathematical proof. NNs, particularly deep and non-linear ones, are resistant to conventional proof methods. It is extremely challenging to formally prove that a trained NN exactly realizes the target cryptographic function over its whole input space and is secure against all conceivable classical, quantum, and AI-based attacks. This dependence on empirical verification instead of formal proof is a significant obstacle to adoption in high-assurance systems.

III. Explainability (XAI) and trust: The "black-box" property of NNs is inherent to the issue. Why did the NN properly or improperly decrypt a particular ciphertext? Can we believe that it hasn't internalized unintended hacks or correlations? This opacity stifles debugging, third-party independent security audits, standards certification (Such as FIPS or common criteria), compliance certification, regulatory clearance, and most importantly, user and organizational trust.

IV. Securing the AI crypto model (MLSecOps): Hardened defense needs protecting the AI model through its lifecycle – securing against adversarial inputs (e.g., via adversarial training), shielding parameters from extraction (e.g., model watermarking, secure enclaves), detecting data poisoning, and guaranteeing secure deployment and updates. This requires expert skills and equipment usually distinct from typical IT security.

V. Scalability, performance, and integration: Running heavy NN models requires huge computational capacities, memory, and power consumption, typically contradicting target environment constraints (Embedded systems, IoT devices). It usually requires hardware acceleration, bringing cost and complication. Scaling those solutions across a heterogeneous network safely, with performance considerations, is difficult when compared to standard, optimized algorithms for classical versions. The latency of the inference of the NN needs to be tolerable, depending on the target application (Real-time communication vs. Offline encryption).

VI. Understanding complex laws: AI and data protection laws like the GDPR and the EU AI Act are continually changing and might differ from one region to another. AI-based crypto systems must comply with rules that safeguard individuals' rights over their data, prevent bias, ensure transparency of algorithms, and satisfy high-security standards. Demonstrating that advanced NN models comply with these rules can prove difficult.

VII. Creating a safe environment: AI and ML systems demand the adoption of secure software development practices. A safe environment involves safeguarding data during training, validating models extensively with security in mind, versioning models as well as data, and monitoring for changes or attacks once models are deployed.

VIII. Adapting to fast changes: AI technology is rapidly advancing, affecting both attack methods and defense strategies. New vulnerabilities or attack methods are found more quickly than before. Cryptographic systems involving AI need to be flexible and regularly updated to keep up with AI-driven threats. Keeping ahead requires a strong, ongoing effort in threat intelligence and security maintenance.

5 | Discussion and Future Directions

5.1 | Synthesis: A Deeply Entwined Co-Evolutionary Intertwining

The arguments put forth synopsize to one firm conclusion: AI and contemporary cryptography are irretrievably entwined, locked in a cycle of co-evolution where breakthroughs and setbacks in one stimulate advances in the other. This partnership is more than a mere tool user. AI stands as a compelling accelerator of cryptographic systems, offering new ways to maximize algorithms (Such as the autonomous creation of secure S-boxes discussed by [7] and simplify security processes. At the same time, AI is an equally compelling threat, giving attackers capabilities that significantly strengthen cryptanalysis functionality, most notably seen in sophisticated SCA methods [8], and potentially exposing weaknesses sooner than conventional methods. In addition to this direct engagement, AI serves as a key enabler for overall cybersecurity, fueling threat intelligence, anomaly detection, and privacy management [10] – frequently complementing cryptographic defenses.

Most importantly for the modern age, AI systems themselves constitute a new, high-value asset class in dire need of protection [1]. These systems handle unprecedented amounts of potentially sensitive information and make-or-break decisions, and their integrity and confidentiality are therefore the most important considerations. The pace of development in AI guarantees this symbiotic and sometimes antagonistic connection will only become more intense, bringing computational intelligence and security mechanisms ever closer together and requiring coordinated approaches to navigating the future. This co-evolution requires ongoing accommodation; static protection is inadequate if both cryptographic ecosystems and AI attack means dynamically change.

5.2 | Key Trends Shaping the Landscape

The trajectory of this AI-cryptography convergence is being actively shaped by several interconnected and accelerating trends:

I. The urgent need to secure AI systems: The debate has firmly turned from the issue of using AI for security to the necessity of securing AI. As noted by the ISACA Journal [1], inherent vulnerabilities of AI models – being vulnerable to data poisoning during training, adversarial evasion at inference, model theft potential, and biases embedded within – pose strong organizational risks. Compromised AI can lead not only to data breaches (Exposing PII/PHI utilized in training or proprietary model details) but also to disastrous failures in essential infrastructure, tampered financial forecasts, or biased autonomous choices [13]. Therefore, building trust in AI requires integrating strong security and privacy measures across the full AI lifecycle, pushing AI security from a specialized concern into a fundamental requirement for uptake.

II. Maturation and diversification of PETs for AI: Cryptography PETs like HE, FL, SMPC, and ZKPs are also evolving rapidly. These are required to maintain the protection of AI data. HE provides the ability to compute on encrypted data, and thus maintain privacy in cloud AI systems and collaborative data analysis. FL is helpful since it trains models without centralizing sensitive data, which is highly critical for applications such as healthcare. Though challenges such as slower performance and user complexity exist, research continues to mitigate these. These involve optimizing algorithms, building better hardware such as dedicated chips for FHE, and combining various PETs. Researchers are also working towards developing solutions optimized for particular AI designs. These initiatives reflect growing potential for applying these privacy-protecting technologies to actual AI use cases.

III. The pervasive and Dual-use impact of generative AI: Generative AI, including Large Language Models (LLMs) and other deep generative models, discussed at events like Cypher 2024 [12], represents a transformative force with profound security implications. Positively, generative models can synthesize realistic data to augment scarce training sets for security classifiers, create diverse "digital twin" environments for testing defenses, or even assist developers by generating potentially more secure code snippets. Negatively, their capabilities are potent tools for attackers [2]. They enable hyper-realistic phishing emails and deepfake generation at scale, automate the creation of polymorphic malware that evades signature-based detection, and can probe systems for vulnerabilities or generate novel attack vectors targeting both traditional systems and other AI models. Detecting and defending against AI-generated threats, while responsibly harnessing generative AI for defense, is a defining security challenge.

IV. Sustained focus on effective deployment, integration, and security for MLOps: The emphasis is now shifting from demonstrating potential to achieving effective, long-lasting deployment [5]. The effective deployment includes smooth integration of AI security tools into current security infrastructure (SIEMs, SOAR platforms) and development processes (DevSecOps pipelines evolving into SecMLOps pipelines). Top of mind are managing the performance overhead of security on AI applications, ensuring that the different platforms (Cloud, edge, hybrid) play nicely together, developing standardized APIs, and fostering the specialized expertise needed to operate secure AI operations. Success is now defined in terms of real, measurable enhancements in security posture, resiliency, and efficiency, not theoretical capability.

V. Post-quantum transition in the shadow of AI: The transition to PQC algorithms is a gigantic cryptographic endeavor that crosses heavily with the development of AI. AI methods can support the transition by contributing to the serious security examination of newly proposed PQC candidates, potentially detecting subtle statistical vulnerabilities or implementation bugs more quickly than traditional techniques [7]. Yet, implementing and operating PQC within systems that already have sophisticated AI components adds substantial architectural and operational complexity. Additionally, there's a possible risk that AI itself might facilitate faster breaking of early or less-than-perfect PQC deployments once large-scale quantum computers are available, calling for extraordinary rigor in PQC design and verification.

VI. The requirement for AI-powered, proactive threat modelling: Threat modelling must shift from the traditional adversary suppositions to proactive inclusion of AI capacity, security experts say [2]. The question now is: How could an attacker with AI technology target our specific data, systems, or AI models? What new attack vectors does AI present? Threat modelling necessitates continuous intelligence collection on offensive AI techniques and involves constructing defenses anticipating AI-powered probes, evasions, and exploits that may be able to enable such an effort by simulating powerful AI attackers.

Table 3. Key Trends Shaping the Artificial Intelligence-Cryptography Landscape.

Trend	Key Implication / Challenge
Securing AI systems	Addressing inherent AI model vulnerabilities (Poisoning, evasion) is critical.
Maturation of PETs for AI	PETs (HE, FL) advance AI privacy but face performance and complexity hurdles.
Pervasive impact of generative AI	Generative AI poses a dual-use threat/opportunity (Attacks vs. defense aid)
Focus on secure MLOps/AISecOps	Practical, secure AI deployment (MLOps) faces integration and skill gaps.
PQC transition and AI interaction	AI aids PQC analysis; integration is complex and potentially adds risk.
Need for AI-powered proactive threat modelling.	Threat models must proactively incorporate AI attacker capabilities.

5.3 | Emerging Research Areas and Future Needs

Addressing the complexities of this interconnected landscape requires focused research efforts across multiple frontiers:

I. To formally verify AI-designed cryptography, it is crucial to close the gap between the innovative potential of AI in creating new cryptographic primitives and the absolute need to guarantee mathematical security. Research frontiers include developing frameworks for providing bounded security assurances for AI-designed components, working on AI systems that are specifically limited to produce provably secure constructs, or working on hybrid methodologies that use AI in discovery with corresponding formal methods or interactive theorem provers for proving.

II. Reliable AI for security operations (Explainability and robustness): Robustness (Adversarial manipulation, data poisoning, and concept drift) and explainability (XAI) are necessary for AI systems that assess threats or make security decisions. For verification, debugging, compliance, and sound incident response, research should aim at building naturally robust AI architectures, good adversarial training methods suitable for security data, and practical XAI methods (Such as modifying SHAP, LIME, or attention mechanisms).

III. Scalable and high-performance PETs for advanced AI: Large-scale deployment of HE, FL, SMPC, etc., will depend on transcending performance limits. The deployment implies innovation in optimized cryptographic algorithms for particular AI computation (e.g., matrix-vector multiplication, non-linear activation operations), co-designing hardware accelerators, optimization of communication protocol for distributed solutions (FL/SMPC), and designing library-friendly implementations that abstract cryptographic detail from AI authors. Hybrid PET schemes integrating the advantages of various techniques for the best possible trade-offs are also significant areas of investigation.

IV. Robust defenses against AI-based attacks: To counter an attacker who uses AI, a proactive "defense-in-depth" strategy is necessary. AI-based attacks entail research and development in: 1) AI-based detection of malicious AI-generated artifacts (Code, deepfakes, and disinformation), 2) hardening systems and protocols to be inherently more resistant to AI-driven analysis and probing, and 3) AI systems that are made to recognize and thwart adversarial AI techniques in real-time, moving toward a defensive stance against AI vs. AI.

V. Standardization for secure AI ecosystems: It is necessary to collaborate to create broad industry-adopted standards and best practices [3]. Standardization includes secure AI data management (Provenance, integrity, privacy), resilient model validation and test procedures (Adversarial robustness tests), secure API design for AI services, best practices for secure composition of AI and cryptographic components, and complete secure AI lifecycle management frameworks (e.g., NIST AI RMF extensions).

VI. Examining AI in security auditing: We are investigating how AI can speed up security audits in intricate networks that use cryptography. AI can help by spotting changes in security settings, making sure regulations are followed, spotting flaws in cryptographic code [13], and carrying out real-time checks to maintain order. These updates improve the protection of important data and reduce errors. AI can streamline the entire procedure and update and secure systems.

VII. Optimizing quantum security with AI: We are looking at expanding AI's role beyond analyzing PQC to include real-time improvements for QKD networks. Optimizing quantum security with AI might mean adjusting error correction methods or key-sharing rates based on AI's predictions about problems like channel noise or attempts to eavesdrop. AI can also help in planning and confirming complex systems that resist quantum threats and potentially explore new cryptographic techniques that mix classical and quantum methods, utilizing insights derived from AI.

5.4 | The Imperative of a Holistic, Responsible Approach

Technology by itself is not enough to provide security and trust for AI and cryptography. A good framework with the proper practices, good governance, ethics, and uniform regulations needs to be enforced. Here is how it is possible:

I. Secure development practices: Secure development lifecycles are to be followed by companies in AI, named SecMLOps. Security reviews, scanning of code and AI model vulnerability, thorough testing like adversarial testing, and secure initial installation are part of it.

II. Strong governance: There must be clearly defined policies for handling AI data. Handling AI data includes data collection, labeling, storage, and retention. It's important to monitor the source and updates of AI models, manage access to models and data, take responsibility for decisions made by AI, and have robust plans to deal with AI failures or breaches.

III. Important ethical considerations: When creating AI systems, particularly for security, it's vital to focus on ethics. We need to look closely at algorithms to catch and fix any biases so that results are fair for all. It's also key to keep these systems clear and understandable by explaining how they work. Additionally, we need to set clear rules on using AI in security to make sure we protect people's privacy while keeping them safe. Balancing these aspects is key to using AI responsibly.

IV. Keeping up with regulations: Companies need to be aware of global rules about AI and data protection, like GDPR and CCPA. Their AI and cryptography setups should meet current and upcoming standards for compliance.

By combining tech innovation with solid engineering, smart governance, ethical choices, and following regulations, AI and cryptography can help create a secure and reliable digital future.

6 | Conclusion

The intersection of cryptography and AI is a paradigm shift in the digital security landscape, combining a very complex, interwoven, and co-evolutionary relationship. This discussion has shown the multi-faceted nature of this dynamic:

AI is a good catalyst, offering revolutionary potential to enhance cryptographic systems via advanced design optimization (e.g., designing improved S-Boxes), algorithmic efficiency improvements (e.g., optimizing QKD parameters), and optimizing security operations via intelligent automation and anomaly detection. Yet this integration also brings with it substantial security challenges. AI greatly enhances the cryptanalysis tools, especially via sophisticated SCA with the ability to reveal secrets from physical implementations, and spurs a new breed of adaptive, AI-based cyber threats from hostile forces aiming to subvert security mechanisms and take advantage of system vulnerabilities.

Critically, the review reveals that securing the AI models themselves is now a critical requirement for trustworthy digital systems [1]. As AI becomes central to critical operations, its vulnerabilities—to data poisoning, adversarial evasion, model theft, and intrinsic biases—form a large attack surface that must be robustly addressed. Cryptographic methods, ironically, turn out to be crucial for AI security, where HE, FL, SMPC, and possibly ZKPs provide ways to accomplish privacy-preserving computation and cooperative AI without exposing sensitive information.

Thus, future development in this field depends not only on utilizing the strengths of AI but on assiduously overcoming its built-in vulnerabilities and guaranteeing its proper use. Developing trustworthy AI-fortified cryptographic solutions calls for a multidisciplinary approach. It needs to be accompanied first by ongoing innovation in creating robust and resilient AI models hardened specifically against recognized attacks, such as adversarial manipulation and data poisoning. Second, ensuring trust requires profound strides in explainability (XAI) to enable security professionals and auditors to comprehend and validate the rationale behind AI-informed security decisions. Third, the potential for quantum computing presents an imminent threat and demands the swift development, rigorous testing, and eventual deployment of quantum-resistant cryptographic solutions (PQC), a challenge that can perhaps be facilitated by AI but one that also brings new challenges into system designs. Fourth, bringing secure systems to life requires the inherent integration of security across the whole AI development process (MLSecOps/AISecDevOps) away from reactive patching towards proactive, security-by-design methodologies encompassing data management, model training, validation, deployment, and continuous monitoring [3]. Lastly, navigating the intricate subtleties at the nexus of AI and cryptography requires ongoing and diligent interdisciplinary collaboration. Progress entails the overlap of expertise in cryptography, ML, systems security, hardware engineering, quantum physics, formal methods, ethics, and policy governance.

By promoting such cooperation, we can effectively unlock the deep power of AI to construct much more adaptive, effective, and smarter cryptographic defenses at the same time that we enact the necessary controls, cryptographic and otherwise, so that the AI elements themselves are secure and trustworthy. The high aim has to be achieving a conscious balance between taking advantage of rapid innovation and ensuring robust security, user privacy, and ethical alignment, thereby unlocking the possibilities for a more secure, resilient, and trusted digital future in the more intelligent age.

Acknowledgments

We would like to express our sincere gratitude to several individuals and groups who contributed to the successful completion of this research. First, we thank the Department of Computer Science at Pranveer Singh Institute of Technology for providing the resources and environment that facilitated this study. We are particularly grateful to Prof. Dr. Pradeep Kumar Singh Sir and Vishal Chaubey Sir for their valuable insights and feedback during the conceptualization, validation, writing, and review phases of this project. Their expertise significantly enhanced the quality of our work.

Author Contribution

This research was a collaborative effort, and the contributions of each author are outlined below:

Conceptualization: Archit Awasthi and Akhand Pratap Shukla. Both authors contributed equally to the initial idea, research question formulation, and overall scope of the study.

Methodology: Archit Awasthi and Akhand Pratap Shukla. The methodology, including the selection of relevant literature, identification of data sources, and the overall analytical approach, was developed jointly.

Literature review: Archit Awasthi. Data Curation: Akhand Pratap Shukla. Writing – Original Draft Preparation: Archit Awasthi. Archit Awasthi took the lead in drafting the initial manuscript, including the introduction, background, and core applications sections.

Writing – Review & Editing: Akhand Pratap Shukla. Akhand Pratap Shukla provided substantial revisions, focusing on the challenges, future directions, and conclusion sections, and ensured the overall coherence and accuracy of the manuscript.

Supervision: Dr. Pradeep Kumar Singh and Mr. Vishal Chaubey. Dr. Pradeep Kumar Singh and Mr. Vishal Chaubey provided guidance and oversight throughout the research process. They offered valuable feedback on the research design, methodology, and manuscript drafts. Their expertise significantly improved the quality of the research. They also provided mentorship, assisting with problem-solving and ensuring the project stayed on track.

Project Administration: Archit Awasthi and Akhand Pratap Shukla. Both authors shared responsibilities for managing the project timeline, coordinating tasks, and ensuring progress.

Conflicts of Interest

The authors declare no conflict of interest. This research was conducted independently, without any external influence or funding.

References

[1] VERMA, S. (2024). Securing the future: Mitigating data security concerns in artificial intelligence models. *Information systems audit and control association journal*, *1*, 23–26. https://B2n.ir/pz3346

[2] Zugec, M. (2024). *2024 cybersecurity predictions for AI: A technical deep dive*. https://B2n.ir/hs7319

[3] Amor, G. B. (2024). *The intersection of AI development and security in 2024*. https://aijourn.com/the-intersection-of-ai-development-and-security-in-2024/

[4] Hu, Y., Kuang, W., Qin, Z., Li, K., Zhang, J., Gao, Y., ... & Li, K. (2021). Artificial intelligence security: Threats and countermeasures. *Association for computing machinery computing surveys (CSUR)*, *55*(1), 1-36. https://doi.org/10.1145/3487890

[5] Texploration & Strategic Patenting. (2024). *Innovations in cryptography and security using AI*. https://texploration.blog/2024/07/22/innovations-in-cryptography-and-security-using-ai/

[6] Kshetri, N., Rahman, M. M., Rana, M. M., Osama, O. F., & Hutson, J. (2024). *algoTRIC: Symmetric and asymmetric encryption algorithms for cryptography-a comparative analysis in AI era*. https://doi.org/10.48550/arXiv.2412.15237

[7] Radanliev, P. (2024). Artificial intelligence and quantum cryptography. *Journal of analytical science and technology*, *15*(1), 4. https://doi.org/10.1186/s40543-024-00416-6

[8] Panoff, M., Yu, H., Shan, H., & Jin, Y. (2022). A review and comparison of AI-enhanced side channel analysis. *Association for computing machinery journal on emerging technologies in computing systems (JETC)*, *18*(3), 1–20. https://doi.org/10.1145/3517810

[9] Chen, A. C. H. (2023). Post-quantum cryptography neural network. *2023 international conference on smart systems for applications in electrical sciences (ICSSES)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICSSES58299.2023.10201083

[10] Walia, K., & Mahalingam, K. (2024). *Leveraging artificial intelligence for enhancing security and privacy in modern computing systems*. https://B2n.ir/pk7950

[11] Siegmann, B. S. (2024). *The 2024 year in review: Cybersecurity, AI, and privacy developments*. https://B2n.ir/je8711

[12] Meghna. (2024). *Evolutionary journey of deep generative models: Insights from cypher 2024*. https://B2n.ir/yh5888

[13] Ishtaiwi, A., Al Khaldy, M. A., Al-Qerem, A., Aldweesh, A., & Almomani, A. (2024). Artificial intelligence in cryptographic evolution: Bridging the future of security. In *Innovations in modern cryptography* (pp. 31–54). IGI Global. https://doi.org/10.4018/979-8-3693-5330-1.ch002