Machine learning-based lightweight block ciphers for resource-constrained internet of things networks: a review

Mahendra Shridhar Naik[1], Madhavi Mallam[2], Chaitra Soppinhalli Nataraju[3]
[1]Department of Electronics and Communication Engineering, New Horizon College of Engineering, Karnataka, India
[2]Department of Electronics and Communication Engineering, PES Institute of Technology and Management, Karnataka, India
[3]Department of Electronics and Communication Engineering, GM Institute of Technology, Karnataka, India

Article Info

Article history:

Received Oct 10, 2023
Revised Jan 24, 2024
Accepted Jan 25, 2024

Keywords:

Block ciphers
Cryptanalysis
Internet of things
Lightweight cryptography
Machine learning

ABSTRACT

The increasing number of internet of things (IoT) devices, wearable technologies, and embedded systems has experienced a significant increase in recent years. This surge has brought attention to the necessity for cryptographic algorithms that are lightweight and capable of providing security in resource-constrained environments. The primary objective of lightweight block ciphers is to provide encryption capabilities with minimal computational overhead and decreased power consumption. As a result, they are particularly well-suited for use on devices that have limited resources. At the same time, machine learning methodologies have evolved into powerful mechanisms for the purposes of prediction, categorization, and system optimization. This study introduces a challenges and issues involved in integrating machine learning techniques with the development of lightweight block ciphers.

Corresponding Author:

Mahendra Shridhar Naik
Department of Electronics and Communication Engineering, New Horizon College of Engineering
Kaadubeesanahalli, Bengaluru 560037, India
Email: mahendrasnaik@gmail.com

1. INTRODUCTION

The rapid increase in the number of devices possessing diverse computational capabilities has had a profound impact on the digital environment in contemporary times. Low-constrained devices, which are frequently distinguished by their restricted computing capacity, memory, storage, and energy resources, are at the vanguard of this transformation [1]. These devices are integral elements of the internet of things (IoT), a conceptual framework in which tangible equipment, ranging from common household items to industrial machinery, are interconnected with the internet. This interconnection enables the provision of improved functions, automation, and intelligent decision-making capabilities [2]. The prevalence of low-constrained devices has significantly increased in a wide range of applications owing to their affordability, mobility, and capability to function effectively in conditions where conventional computing devices may not be practical [3]. Illustrative instances encompass wearable health monitoring, intelligent thermostats, and wireless sensor nodes deployed in agricultural areas. Nevertheless, the constrained resources of the aforementioned entities present obstacles in terms of performance, security, and optimal power consumption [4]. The IoT ecosystem leverages the functionalities of these devices, establishing an extensive interconnected network capable of generating, transmitting, and processing data. This enables the development of intelligent urban environments, greater healthcare surveillance, sophisticated industrial automation, and elevated user experiences across several domains. However, the increased interconnection also gives rise to apprehensions over security, privacy, and interoperability [5]. The increasing number of IoT devices and the expanding network of interconnected systems demand the implementation of security measures that are both effective and efficient. Not all gadgets within this ecosystem possess the privilege of ample computing power. Numerous embedded technologies, such as radio frequency identification (RFID) tags and smart cards, exhibit limitations in power, memory, and computing capacities. The aforementioned circumstances have spurred the advancement and acceptance of cryptographic algorithms that are designed expressly for lightweight contexts.

Lightweight block ciphers are a specific category of lightweight cryptographic primitives that aim to provide the same functionality as conventional block ciphers while requiring fewer resources and performing more efficient operations. The objective is to optimize hardware and software requirements while maintaining a high level of security. The challenge at hand is complex, as attaining an ideal equilibrium between efficiency and security is not easily accomplished.

The present lightweight block cipher, which was introduced by Bogdanov *et al.* in 2007 [6], stands out as one of the first and most prominent examples in its field. With a primary emphasis on achieving minimum hardware implementation, this cipher established a pattern for numerous subsequent lightweight ciphers. Simon and Speck are notable lightweight block ciphers that were developed by the U.S. National Security Agency (NSA) [7]. These algorithms were specifically designed to provide efficient encryption capabilities in both hardware and software implementations. The significance of comprehending the features, applications, and limitations of these ciphers becomes increasingly crucial as their application fields continue to broaden. This further emphasizes the continuous requirement for study in order to determine the resilience of these systems against different cryptographic attacks and enhance their implementation over a wide range of platforms. However, the process of designing, optimizing, and validating these lightweight cryptographic solutions might present significant challenges. Machine learning assumes a crucial function in this context. The utilization of machine learning has the potential to facilitate the identification and development of novel cryptographic methods. Machine learning models could improve performance and security in situations where resources are limited by using existing cryptographic primitives and attacks to create new structures or change old ones [8]. Machine learning approaches can be utilized for the purpose of detecting weaknesses in cryptographic algorithms. Machine learning models have the capability to undergo training in order to make predictions regarding potential vulnerabilities in cipher implementations or to identify instances of side-channel leakages. This ability to provide meaningful feedback to cryptographers is well documented [9]. The utilization of machine learning can facilitate the real-time adaptation of cryptographic systems by leveraging its predictive capabilities. Optimizing resource utilization is particularly critical for low-constrained devices. For example, a machine learning-driven system has the capability to adapt cryptographic parameters by considering the identified threat level or the resources that are currently accessible [10]. Machine learning has the potential to enhance the efficiency of hardware or software implementation of lightweight cryptographic algorithms on specific device architectures. Machine learning models have the ability to propose implementation techniques that optimize efficiency by examining device-specific limitations and performance indicators [11]. Naik *et al.* [12] compares various block code suitable for resource constrained network. In brief, the objective of lightweight cryptography is to offer secure solutions for devices with limited resources. In this context, machine learning serves as a supplementary tool, facilitating the enhancement, verification, and flexibility of these cryptographic systems.

2. RELATED WORKS

This section provides an overview of the several lightweight block ciphers that are appropriate for use in low-constrained IoT devices and networks. Banik *et al.* [13] presents GIFT, a novel lightweight block cipher that is presented as an evolution or alternative to the PRESENT cipher. GIFT's design principles likely prioritize maximum security while minimizing resource consumption, particularly for devices with constrained computational capacity, memory, and energy. The authors will evaluate GIFT's effectiveness in terms of encryption speed, memory footprint, power consumption, and hardware implementation costs. These metrics are indispensable for assessing the usability of any lightweight encryption algorithm. A significant portion of the paper would be devoted to a comprehensive evaluation of GIFT's security. Given the increasing emphasis on physical security for lightweight ciphers, this would entail evaluating their resistance to known cryptographic attacks and possibly side-channel attacks. Dinu *et al.* [14] introduces SPARX, a lightweight, side-channel-resistant block cipher. Side-channel attacks employ a cryptosystem's design, such as its power consumption or electromagnetic wave leakage, rather than algorithmic vulnerabilities. SPARX mitigates side-channel vulnerabilities using a unique architecture. This is crucial for resource-constrained devices without advanced protections. A full performance investigation of SPARX, comparing its encryption speed, power consumption, and resource utilization with competing lightweight block ciphers, is likely. A thorough security examination would be the paper's focus. The writers would test the cipher's resistance against side-channel and cryptographic attacks.

Zhang *et al.* [15] offers RECTANGLE, a lightweight block cipher for software and hardware implementations that prioritizes bit-slice efficiency. A typical block cipher design is RECTANGLE's substitution-permutation network (SPN). The cipher uses 64-bit blocks. RECTANGLE offers 80 or 128-bit keys. RECTANGLE's bit-sliced implementation allows concurrent execution of many ciphers, making it stand out. The setup improves throughput, especially in software implementations. The authors evaluated RECTANGLE's security against differential and linear cryptanalysis. These assaults are resistant to encryption, ensuring its security for intended purposes. Designed for efficiency across platforms, RECTANGLE. Its design balances hardware and software performance. The bit-slice technique allows software to parallelize encryption and decryption operations, enhancing throughput. The cipher is versatile for diverse use cases since it supports lightweight cryptographic applications on multiple platforms. Its balanced performance measurements make it suited for software-centric and hardware-restricted systems. The strong, lightweight block cipher RECTANGLE is tailored for numerous platforms. Its design, notably its emphasis on bit-sliced implementation, makes it a unique solution for software and hardware cryptography demands.

Prince [16], a lightweight block cipher for low-latency applications, ensures fast encryption and decryption in pervasive computing environments. Prince uses a unique mirrored structure to quickly encrypt and decrypt by recycling most of the encryption process. Prince uses 64-bit blocks. The cipher uses a 128-bit key split into two 64-bit halves for ciphering and key whitening. Prince's ultra-low latency makes it ideal for applications requiring near-instantaneous encryption or decryption. Prince resists popular cryptographic assaults, and the authors include security ratings against cryptanalysis. The “alpha reflection” attribute ensures that the decryption process reflects the encryption processes, which reduces latency and affects security. Hardware versions of the cipher work well, consuming little space and power. Its design is optimized for pervasive computing environments that require fast computations. Prince excels in pervasive computing, where devices are seamlessly integrated into common objects and require real-time or near-real-time computation. IoT sensors, automotive systems, and other devices that need low-latency encryption and decryption can use it. Low-latency operations are Prince's new approach to block cipher architecture. It adds value to lightweight cryptography by targeting the fast-changing world of pervasive computing, where instantaneous cryptographic operations are becoming essential.

LED [17] uses substitution-permutation network (SPN). Key size is 64 bits, and block sizes are 64 bits (LED-64) and 128 bits (LED-128). LED reduces software and hardware complexity. Sharing components, decreasing operations, and simplifying operations achieve this. LED employs 48 or 64 rounds for security. Each cycle uses S-boxes for substitution and permutation. The authors examine LED's resilience to differential and linear cryptanalysis. A sufficient number of rounds are chosen for security. The architecture ensures diffusion and avalanche effects, so changing one input bit alters several output bits. The authors describe LED hardware implementation and performance measurements. The area-optimized hardware is suited for resource-constrained devices. The speed-code size trade-offs of software solutions are also considered. Lightweight block cipher LED is for resource-constrained situations. It balances security and efficiency, making it suited for IoT and embedded systems.

In this study, Al Tawy and Youssef [18] provide an innovative methodology that utilizes generative adversarial networks (GANs) for the autonomous generation of cryptographic algorithms. The model, known as CryptoGAN, undergoes repeated development of cryptographic techniques with the objective of creating schemes that possess both lightweight characteristics and robust security. The method described has considerable promise for low-constrained devices due to its ability to automatically customize encryption techniques according to the limitations of the device.

Bao *et al.* [19] explores the convergence of neural networks and cryptanalysis, with a specific emphasis on the capacity of machine learning algorithms to decrypt encrypted data. They demonstrate the application of neural networks in identifying vulnerabilities in cryptographic methodologies. This ensures the security of lightweight cryptographic systems, particularly for devices that possess constrained resources. In light of the increasing concern regarding the potential risks posed by cyberattacks utilizing machine learning techniques, the researchers in reference [20] investigate the development of cryptographic approaches that may effectively withstand these attacks. The significance of their research lies in its contribution to the comprehension of the interplay between emerging cyber threats and the cryptographic countermeasures required for devices with limited resources. In their article, Hill and Bellekens [21] emphasize the importance of selecting optimal parameters in the field of cryptography. Machine learning models are employed to identify the optimal cryptography parameters for each given task. This ensures that lightweight cryptography for devices with constrained resources is optimized in terms of both security and efficiency. Rana and Mamun [22], focus their research on the domain of the IoT and highlight the importance of implementing cryptographic solutions that are energy-efficient. The machine learning model developed by the researchers focuses on symmetric encryption techniques that have been improved to prioritize power conservation, which is crucial for IoT devices that rely on battery power. The incorporation of reinforcement learning into the domain of lightweight block ciphers is explored in a study conducted by Kim *et al.* [23], which presents the concept of dynamic adaptability. The methodology employed by the researchers guarantees that the cryptographic system has the capability to dynamically modify its parameters in response to feedback, improving both its performance and security. With a focus on wearable technologies, the study by Hamza and Minh-Son [24] explores the use of deep learning in the creation of adaptive cryptosystems. The proposed methodology guarantees the security of wearable devices by employing context-aware encryption techniques that take into account the distinct limitations and user-specific data associated with such devices. The authors of reference [25] came up with a way to use convolutional neural networks (CNNs) to improve the key generation process in lightweight cryptographic systems. The results of their study demonstrate notable enhancements in efficiency, particularly advantageous for devices that possess constrained processing capabilities. The researchers in reference [26] propose a novel methodology that incorporates transfer learning into lightweight cryptographic techniques. Through this approach, researchers are able to integrate the advantages of conventional cryptographic methods with the advancements provided by machine learning, leading to the development of encryption approaches that are both more resilient and efficient for devices with limited resources. The authors [27] employ machine learning techniques to identify possible security risks in IoT devices, recognizing the inherent vulnerabilities associated with side-channel attacks. The research highlights the crucial significance of machine learning in safeguarding the security of cryptographic implementations inside the continuously expanding IoT environment.

3. MACHINE LEARNING IN CRYPTOGRAPHY

The subjects of machine learning and cryptography are two that are connected in many ways and have many applications [28]. Machine learning is a subfield of artificial intelligence that makes use of algorithms and data in order to learn from previous experiences and carry out tasks without the need for explicit programming. The field of study known as cryptography focuses on the development and evaluation of technologies that ensure private computations and communications. Depending on the desired outcome and the nature of the challenge, there are a variety of approaches that can be taken when implementing machine learning in cryptography. Here are some examples of probable outcomes: Developing new cryptographic primitives and protocols with the use of machine learning for instance, one may make advantage of machine learning in order to develop safe pseudorandom functions, hash functions, encryption methods, or digital signatures [29]. The parameters or structures of already existing cryptographic constructs can likewise be improved through the application of machine learning. Applying machine learning techniques in order to evaluate the safety or effectiveness of cryptographic systems for the purpose of doing cryptanalysis, one may, for instance, employ machine learning to perform tasks such as locating flaws, breaking ciphers, or recovering keys [30]. The effectiveness, robustness, or usefulness of cryptographic systems can also be evaluated with the help of machine learning. Increasing the usefulness or safety of cryptographic systems through the application of machine learning techniques for instance, machine learning can be used to offer adaptive security, which involves modifying the level of protection based on factors such as the environment or the threat that is being posed. Machine learning can also be utilized to provide computation that protects users' privacy, and one example of this would be carrying out machine learning tasks on top of encrypted data. It is necessary to have a solid understanding of both machine learning and cryptography, in addition to the tools and methods that are currently at one's disposal. Only then can one successfully integrate machine learning in cryptography. The following is a list of some of the more common tools and techniques:

Models of machine learning include neural networks, decision trees, support vector machines, and k-means clustering, among others. These models can be utilized in the processes of pattern recognition, data prediction, and data classification. Figure 1 is a flowchart illustrating the various uses, problems, and future prospects of machine learning in the field of cryptography. Machine learning is a subdivision of artificial intelligence that encompasses the development of systems capable of acquiring knowledge from data and generating predictions or choices. Cryptography is the discipline concerned with safeguarding information by the utilization of codes, ciphers, or other methodologies. Figure 1 illustrates several potential applications of machine learning in cryptography, including:

Cryptanalysis refers to the application of machine learning techniques to dismantle or evaluate cryptographic systems, with the aim of identifying vulnerabilities, deciphering passwords, or retrieving encryption keys. Enhancing the security and efficiency of communication protocols, such as encrypting messages, compressing data, or validating identities, through the utilization of machine learning. Intrusion detection systems employ machine learning techniques to discover and thwart illegal access or attacks on networks, systems, or devices. This includes the detection of malware, spam, and phishing attempts. Fraud Detection: Employing machine learning techniques to identify and thwart deceitful acts or transactions, such as credit card fraud, identity theft, or money laundering. Predictive analysis refers to the utilization of machine learning techniques to examine and predict trends or patterns in various types of data, including stock market fluctuations, customer actions, and meteorological conditions. Anonymity: employing machine learning techniques to safeguard the privacy and anonymity of users or data, such as concealing IP addresses, obfuscating identities, or producing fabricated data.

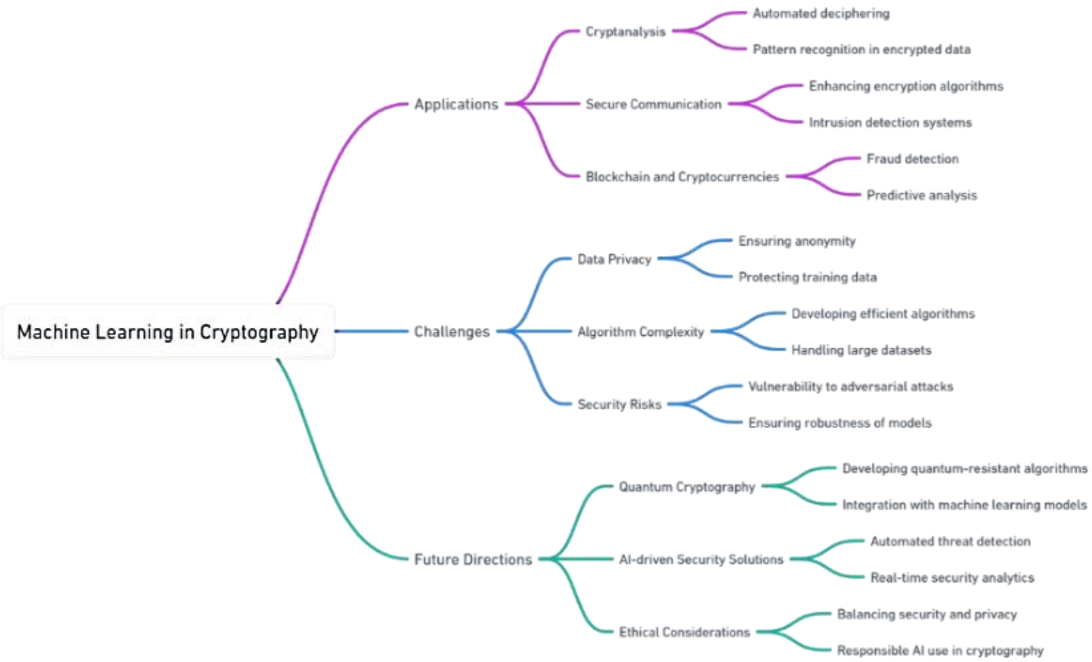
Machine Learning in Cryptography
Applications
Cryptanalysis
Automated deciphering
Pattern recognition in encrypted data
Secure Communication
Enhancing encryption algorithms
Intrusion detection systems
Blockchain and Cryptocurrencies
Fraud detection
Predictive analysis
Challenges
Data Privacy
Ensuring anonymity
Protecting training data
Algorithm Complexity
Developing efficient algorithms
Handling large datasets
Security Risks
Vulnerability to adversarial attacks
Ensuring robustness of models
Future Directions
Quantum Cryptography
Developing quantum-resistant algorithms
Integration with machine learning models
AI-driven Security Solutions
Automated threat detection
Real-time security analytics
Ethical Considerations
Balancing security and privacy
Responsible AI use in cryptography

Figure 1. Relationship between machine learning and cryptography

Challenges: Figure 1 also illustrates certain obstacles or complexities associated with the utilization of machine learning in cryptography, including: Data privacy: The act of safeguarding machine learning data from being stolen, leaked, or misused, which involves measures to protect sensitive information, adhere to legislation, and honor user consent. Algorithm complexity: Addressing the intricacy and scalability of machine learning algorithms, such as diminishing computational expenses, enhancing performance, or preventing overfitting. Managing and processing extensive datasets for machine learning involves tasks such as storing, retrieving, and cleaning data, as well as addressing issues like missing, noisy, or imbalanced data.

Ensuring reliability: Guaranteeing the dependability, resilience, and credibility of machine learning systems by activities such as result verification, error handling, and decision explanation. Security risks: Ensuring the prevention or reduction of security risks or threats associated with machine learning, such as adversarial attacks, backdoors, or spoofing. Figure 1 also indicates potential future applications or prospects of employing machine learning in cryptography, including: Quantum cryptography refers to the utilization of principles from quantum physics to develop cryptographic systems that offer robust security against quantum computers. This includes techniques like quantum key distribution, quantum encryption, and quantum random number generation.

Integrating machine learning models: Employing cryptographic methods, such as homomorphic encryption, safe multiparty computation, or differential privacy, to bolster the security and privacy of machine learning models. Automated threat detection involves the utilization of machine learning algorithms to autonomously identify and react to novel or developing threats or attacks, such as zero-day exploits, ransomware, or botnets. Ethical considerations: Examining the ethical, societal, and legal ramifications of employing machine learning in cryptography, including issues of equity, responsibility, and openness.

Responsible AI use in cryptography involves employing machine learning techniques in a manner that is both responsible and advantageous. This includes adhering to established best practices, standards, and guidelines, as well as including human oversight, control, and intervention. Several examples of algorithms used in machine learning include gradient descent, backpropagation, stochastic gradient descent, and genetic algorithms. These techniques can be utilized to train machine learning models, optimize existing models, or evolve existing models. Frameworks for machine learning include TensorFlow, PyTorch, Keras, and Scikit-learn, among others. Building and deploying machine learning applications is made easier with the help of these frameworks, which can provide high-level APIs and libraries. Primitive forms of cryptography include things like symmetric-key encryption and public-key encryption, as well as digital signatures and hash functions. These fundamental building blocks are capable of delivering essential security services such as non-repudiation, authentication, integrity, and confidentiality. Cryptographic protocols include things like key exchange, secure multiparty computation, zero-knowledge proofs, and homomorphic encryption, amongst others. These protocols are capable of providing more advanced security services, such as key agreement, cooperative computation, verified computation, or encrypted computation, among other possibilities. There are a variety of cryptographic libraries available, including OpenSSL, Crypto++, NaCl, and PyCrypto. These libraries are able to supply low-level application programming interfaces (APIs) and implementations for the utilization of cryptographic primitives and protocols.

4. CRYPTOANALYSIS USING MACHINE LEARNING

The use of machine learning techniques in the study of cryptographic algorithms in order to locate flaws or vulnerabilities is a new and developing field known as cryptoanalysis utilizing machine learning (or simply cryptoanalysis). Using distinguishers, classifiers, or key recovery attacks that make use of some statistical or structural aspects of the cipher, machine learning can be utilized to construct these types of tools. Methods already in use for decryption, such as differential and linear cryptanalysis, may benefit from the application of machine learning to enhance their effectiveness and precision.

Figure 2 illustrates the sequential steps involved in cryptanalysis through the utilization of machine learning. Cryptanalysis refers to the systematic examination and deciphering of cryptographic systems, including codes, ciphers, and encryption techniques. Machine learning is a subdivision of artificial intelligence that focuses on developing systems capable of acquiring knowledge from data and generating predictions or judgments. Figure 2 illustrates many machine learning techniques utilized in cryptanalysis, including: Supervised learning refers to the process of acquiring knowledge from data that has been labeled, and subsequently making predictions based on the patterns that have been learned. Unsupervised learning refers to the process of extracting hidden patterns or groupings from unlabeled data. Deep Learning refers to the utilization of several layers of artificial neural networks to acquire knowledge of intricate and nonlinear characteristics from the given data.

Figure 2 illustrates the sequential process of gathering and organizing data for the purpose of machine learning, encompassing the following stages: Preprocessing is the process of cleaning, altering, or normalizing the data to ensure its suitability for machine learning. Feature extraction involves identifying and isolating pertinent or valuable data attributes that can aid in the process of cryptanalysis. Model training: The process of training the machine learning model using the provided data and fine-tuning the parameters to maximize performance. Data decryption and analysis: Figure 2 illustrates the various activities and obstacles associated with decrypting and analyzing data with machine learning, including: Managing and manipulating enormous volumes of data, which could potentially be encrypted or encoded. Collected encrypted text samples: Acquiring or producing a sufficient number of encrypted or ciphered text examples suitable for machine learning purposes. Issues with the protection of personal information: Ensuring the privacy and confidentiality of both the data and the users involved in cryptanalysis. Figure 2 illustrates some potential applications and advantages of employing machine learning in cryptanalysis, including: Adapting to evolving encryption methods: Employing machine learning to effectively handle the dynamic and advancing encryption methods and approaches. Decrypting ciphered texts: Employing machine learning algorithms to decipher encoded texts and retrieve the original messages or information. Analysis of secure communications: Employing machine learning techniques to scrutinize and oversee secure communications, identifying any irregularities or potential risks. Historical document analysis: Employing machine learning to examine and interpret historical documents that might include encrypted or ciphered texts.

Figure 3 illustrates the procedure of employing machine learning for data encryption and decryption. Data encryption is the procedure of converting data into an incomprehensible format by utilizing a confidential key or algorithm. Data decryption refers to the procedure of recovering the original data from its encrypted state by employing either the same or a distinct key or algorithm. Data encryption: The initial stage involves encrypting the data using an appropriate encryption technique, such as symmetric encryption, asymmetric encryption, or hybrid encryption. Encryption techniques employ mathematical algorithms or processes to obfuscate the data and safeguard it against illegal access or alteration. Obtaining the encrypted data, often known as ciphertext, is the second stage. Ciphertext is the outcome of encrypting data, rendering it typically incomprehensible or devoid of meaning to someone lacking the decryption key or method.

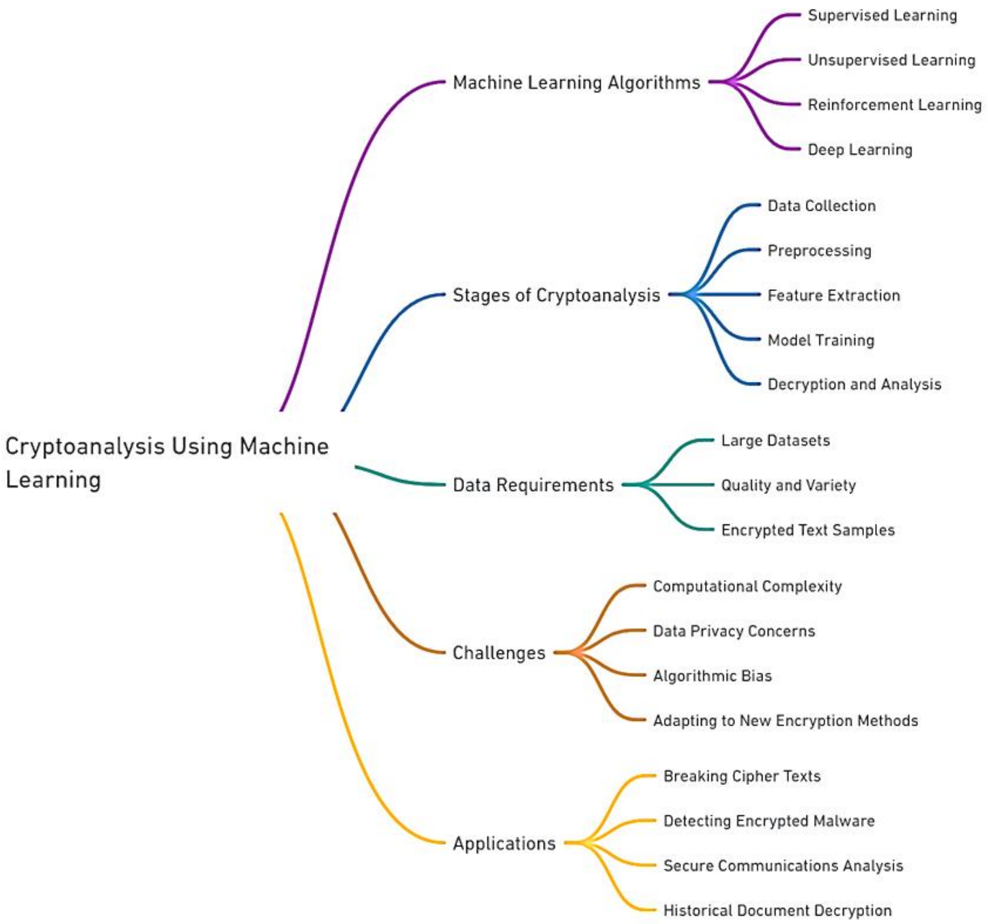

Figure 2. Cryptoanalysis using machine learning

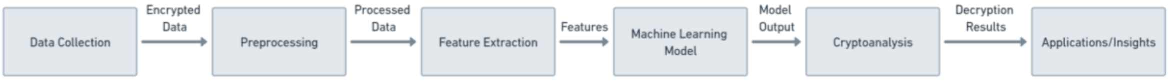

Figure 3. The process of data encryption and decryption using machine learning

Preprocessing: The third stage involves preparing the encrypted data to be compatible with machine learning algorithms. Preprocessing includes the tasks of eliminating noise, errors, or inconsistencies from the data, as well as enhancing its quality, efficiency, or compatibility. Feature extraction: The fourth phase involves extracting distinctive characteristics from the preprocessed data that can aid in cryptanalysis. Features refer to pertinent or valuable information or characteristics that can be extracted from the data, such as frequency, patterns, or statistics. Feature extraction can employ diverse methodologies, including n-grams, histograms, or neural networks. Obtaining the characteristics, also known as inputs or variables, is the fifth stage. Features refer to the data or characteristics that the machine learning model utilizes to acquire knowledge from the data and generate predictions or conclusions. Representation: Select an appropriate machine learning model capable of doing cryptanalysis as the sixth step. A machine learning model is a computational system that acquires knowledge from data and uses this knowledge to generate predictions or make decisions by recognizing patterns. A machine learning model can employ several algorithms, including supervised learning, unsupervised learning, or deep learning. Output from the machine learning model: The seventh phase involves acquiring the output of the machine learning model, sometimes referred to as predictions or judgments. The output is derived by the application of the machine learning model to the features. It has the potential to reveal details about the encrypted material, including the encryption technique, the encryption key, or the plaintext.

Cryptoanalysis refers to the study and analysis of cryptographic systems, with the aim of understanding their weaknesses and vulnerabilities. Step eight involves conducting cryptanalysis by utilizing the results of the machine learning model. Cryptoanalysis refers to the systematic examination and deciphering of cryptographic systems, including codes, ciphers, and encryption techniques. Cryptoanalysis employs a range of approaches, including brute force, statistical analysis, and differential analysis.

Obtaining decryption results or application insights is the ninth and final phase, achieved through the use of cryptoanalysis. The decryption results refer to the restoration of the original data or plaintext from the encrypted data or ciphertext. The application insights refer to the advantages or consequences of decrypting data, such as ensuring secure communication, detecting fraud, or analyzing historical documents.

At CRYPTO'19, Gohr presented one of the most noteworthy works on machine learning-based cryptanalysis. In this work, he employed deep neural net-works to develop a distinguisher for the NSA block cipher SPECK [31] that out-performed the cutting-edge outputs. Gohr's work is one of the most notable works on machine learning-based cryptanalysis. He demonstrated that his neural distinguisher could be included in a key recovery attack that could break 17 rounds of SPECK32/64 with a chosen plaintext complexity of 2^{32} and a time complexity of 2^{32}. However, his study also created some problems regarding the interpretability and explainability of his neural distinguisher since it was not clear how it actually works or what information it exploits. These questions were raised as a result of the fact that it was not clear how it actually worked.

Benamira *et al.* [32] presented at EUROCRYPT 2021, a proposal that offered a comprehensive study and in-depth explanations of the intrinsic workings of Gohr's neural distinguisher as a means of addressing this problem. They showed that the neural distinguisher depends not only on the penultimate and antepenultimate rounds of the algorithm, but also on how the ciphertext pairs are spread out. They also developed a distinguisher that does not rely on neural networks and achieves the same level of accuracy as Gohr's neural distinguisher through the use of pure cryptanalysis. Also, they stripped Gohr's deep neural network down to its most basic parts and showed that, during the learning phase, it builds a good approximation of the cipher's differential distribution table (DDT).

At the conference on Security and Communication Networks 2020, Kim *et al.* [33] presented another piece of work on machine learning-based cryptanalysis. In this study, they suggested a universal deep learning-based cryptanalysis model that locates the key from known plaintext-ciphertext combinations. They demonstrated that their model could recover the key to numerous lightweight block ciphers, such as PRESENT, LED, RECTANGLE, and LILLIPUT, by applying it to these ciphers and demonstrating that it has a high success rate while having a low data complexity. They also compared their model to other models, such as those based on algebra or statistics, and evaluated the advantages and drawbacks of their approach.

Modern cryptography attacks on block ciphers are techniques of determining the secret key or circumventing the encryption algorithm's security. Differential cryptanalysis, linear cryptanalysis, and its combinations, such as differential-linear attacks, miss-in-the-middle attacks, and boomerang attacks, are among the most extensively employed techniques [34]. These attacks take advantage of the statistical aspects or algebraic structures of block ciphers in order to recover the key or narrow the search space. Recently, deep learning-based cryptanalysis has emerged as a new approach, which uses neural networks to learn the relationship between plaintexts, ciphertexts, and keys. This method can be applied to lightweight block ciphers, such as simplified DES, Simon, and Speck, and can recover the key bits when the keyspace is restricted to 64 ASCII characters [35]. Some of the most recent attacks on block ciphers are:

- Side-channel attacks: These attacks exploit physical leakages from cryptographic implementations, such as power consumption or timing variations, to glean information about the internal state of the cipher. Recent advancements in differential power analysis (DPA) techniques, like high-order DPA and template matching with side-channel leakage randomization removal, have demonstrated effectiveness against advanced encryption standard (AES) and other ciphers [36], [37].
- Related-key attacks: These attacks leverage similarities in the key schedules of different ciphers to break one cipher and then attack others. The recent proposal of the Biclique cryptanalysis technique has shown promise in exploiting related-key relationships in some block ciphers, raising concerns about potential vulnerabilities in cipher families [38].
- Algebraic attacks: These attacks utilize algebraic properties of the cipher's S-boxes or round functions to find weaknesses. Recent work on Gröbner basis attacks has targeted ciphers with specific algebraic structures, highlighting the importance of careful S-box design [39].
- Quantum cryptanalysis: While still in its early stages, the emerging field of quantum computing poses a potential threat to the long-term security of classical ciphers. Recent research on quantum Grover's algorithm and its application to specific ciphers underscores the need for investigating quantum-resistant alternatives [40].

5. PROBLEMS WITH USING MACHINE LEARNING FOR CRYPTOGRAPHY AND ANALYSIS

Machine learning is a highly potent methodology that finds application across several domains, encompassing data analysis, pattern identification, classification, prediction, and optimization. But using machine learning techniques with lightweight block ciphers, which are designed for devices with limited resources like IoT sensors, RFID tags, and smart cards, has some problems and tradeoffs. Several obstacles exist in this context. The objective of lightweight block ciphers is to decrease the memory requirements of the encryption and decryption methods, as well as the sizes of the key and state. The reason for this is that low-resource devices possess a restricted memory capacity and are unable to allocate significant amounts of data or code. Nevertheless, it is worth noting that machine learning models frequently necessitate substantial memory resources for the storage of parameters, weights, or characteristics acquired through the training process. Hence, the task of striking a harmonious equilibrium between the memory efficiency of the cipher and the machine learning model presents a formidable challenge.

The primary objective of lightweight block ciphers is to minimize the computational complexity associated with the encryption and decryption techniques, together with the total number of rounds or operations required. The reason for this is that low-resource devices possess constrained processing capabilities and energy consumption, rendering them unable to execute intricate or resource-intensive computations. Nevertheless, it is worth noting that machine learning models frequently necessitate substantial computing complexity for both training and testing processes, particularly in the case of deep learning models, which incorporate numerous layers or nonlinear functions. Hence, a significant problem lies in achieving a harmonious equilibrium between the computing efficiency of the cipher and the machine learning model.

In order to ensure an adequate level of security, lightweight block ciphers need to effectively defend against a range of attack methods, including differential, linear, algebraic, and statistical attacks. The susceptibility of low-resource devices to physical or logical attacks poses a significant risk to the confidentiality, integrity, and availability of both data and equipment. Still, using machine learning models in cipher systems could lead to new vulnerabilities or flaws, such as overfitting, underfitting, adversarial examples, and side-channel attacks, but not limited to these. Hence, an additional problem lies in achieving a harmonious equilibrium between the security level of the encryption and the machine learning model. The implementation of machine learning in lightweight block ciphers for low-resource devices presents several significant obstacles. Additionally, there are several problems pertaining to the design choices, performance measures, assessment techniques, and application situations of both the cipher and the machine learning model.

Machine learning offers intriguing possibilities for cryptography and analysis, but its integration is not without challenges. Here are some key problems to consider: Lack of theoretical guarantees: Unlike traditional cryptographic primitives built on well-established mathematical foundations, machine learning models often lack rigorous theoretical guarantees about their security. This can make it difficult to assess their vulnerability to attacks and prove their long-term resilience [41]. Adversarial vulnerability: machine learning models are susceptible to adversarial attacks, where attackers manipulate data or exploit model biases to compromise their functionality. This poses significant risks in cryptographic applications, where even minor deviations from expected behavior can be catastrophic [42]. Data dependence and bias: machine learning models are heavily dependent on the training data, and biases present in the data can be amplified by the model [43]. This can lead to discriminatory outcomes or inaccurate analysis in security-sensitive contexts. Explainability and transparency: The complex internal workings of many machine learning models make it challenging to understand how they arrive at their decisions. This lack of transparency hinders debugging, raises concerns about fairness, and creates difficulties in verifying their trustworthiness for cryptographic applications [44]. Hardware and resource requirements: Training and deploying complex machine learning models often require significant computational resources and specialized hardware [45]. This can limit their feasibility in constrained environments or resource-sensitive security protocols. Existing research has identified potential vulnerabilities in certain block ciphers, like Grover's algorithm's enhanced attack on PRESENT and Simon's algorithm's impact on certain AES variants. However, these attacks often require significant resources and may not yet pose immediate practical threats [46]. Quantum algorithms, like Grover's, often amplify the effectiveness of existing cryptanalytic techniques like meet-in-the-middle attacks, reducing the required complexity and resources. This combined approach presents a more significant threat than solely relying on quantum algorithms [47]. Cryptographic researchers are actively developing quantum-resistant algorithms and ciphers specifically designed to withstand attacks from quantum computers. Examples include lattice-based cryptography and post-quantum code-based cryptography.

Despite these challenges, the field of machine learning-aided cryptography is actively evolving, seeking to address these limitations and unlock the potential of machine learning for enhancing security and analysis. Hybrid approaches combining traditional cryptographic primitives with machine learning models offer promising avenues for secure and efficient solutions. It's crucial to carefully consider the limitations of machine learning while exploring its applications in cryptography and analysis. By leveraging existing cryptographic expertise and employing rigorous security practices, we can harness the power of machine learning to strengthen security solutions without compromising its core principles.

Machine learning has significant potential for improving security and functionality. Here are some significant advances in machine learning-aided cryptography: Post-quantum cryptography: One of the most significant applications of machine learning lies in developing post-quantum cryptography (PQC) algorithms. These algorithms are designed to resist attacks from quantum computers, which pose a potential threat to classical ciphers. Recent advancements include: Lattice-based cryptography: machine learning is used to design efficient lattice-based cryptosystems, like Kyber and Dilithium, shortlisted for US National Institute of Standards and Technology (NIST)'s PQC standardization [48]. Code-based cryptography: machine learning helps improve the performance and security of code-based cryptosystems like classic McEliece [49]. Key generation and management: machine learning can facilitate secure and efficient key generation and management, essential aspects of cryptographic protocols. Examples include: Machine learning-aided random number generation (RNG): machine learning can generate high quality random numbers for cryptographic applications, addressing concerns about traditional RNGs' vulnerabilities [50]. Threat-aware key management: machine learning models can analyze system activity and threat landscape to dynamically adjust key lifetimes and rotation strategies, enhancing security [51]. Side-channel attack detection and mitigation: machine learning can be employed to detect and mitigate side-channel attacks, which exploit physical leakages like power consumption or timing variations to extract cryptographic secrets. This includes: Anomaly detection using machine learning: machine learning models can learn normal behavior patterns of cryptographic implementations and identify anomalies indicative of potential attacks [52]. Countermeasure design and optimization: machine learning can aid in designing and optimizing countermeasures like masking techniques to thwart side-channel attacks.

6. CONCLUSION

Machine learning and lightweight block ciphers are gaining momentum in the field of IoT security and optimization. Machine learning-assisted analysis can identify vulnerabilities in lightweight block ciphers, leading to more robust ciphers and thwarting side-channel attacks. Machine learning can also optimize lightweight block ciphers implementations, reducing computational overhead and energy consumption while maintaining security levels. This proactive approach offers a significant advantage over static lightweight block ciphers solutions in dynamic IoT environments. However, challenges include ensuring theoretical security guarantees, addressing adversarial resilience, enhancing explainability and transparency, and overcoming hardware and resource constraints. Future directions include joint design of lightweight block ciphers and machine learning algorithms, federated learning for collaborative security, and blockchain-based secure machine learning for authentication and trust.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, “Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm,” *Ad Hoc Networks*, vol. 56, pp. 122–140, Mar. 2017, doi: 10.1016/j.adhoc.2016.12.004.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (IoT): a vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, doi: 10.1016/j.future.2013.01.010.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: a survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, Oct. 2015, doi: 10.1109/COMST.2015.2444095.

[4] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in internet of things: the road ahead,” *Computer Networks*, vol. 76, pp. 146–164, Jan. 2015, doi: 10.1016/j.comnet.2014.11.008.

[5] I. Makhdoom, M. Abolhasani, J. Lipman, R. P. Liu, and J. Hu, “The evolution of IoT attacks and defences: a review,” *IEEE Access*, vol. 9, pp. 31703–31725, 2021, doi: 10.1109/ACCESS.2021.3091053.

[6] A. Bogdanov *et al.*, “PRESENT: An Ultra-Lightweight Block Cipher,” in *Cryptographic Hardware and Embedded Systems - CHES 2007*, vol. 4727, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466.

[7] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, “The SIMON and SPECK lightweight block cIPhers,” in *Proceedings - Design Automation Conference*, Jun. 2015, vol. 2015-July, pp. 1–6, doi: 10.1145/2744769.2747946.

[8] I. K. Dutta, B. Ghosh, A. Carlson, M. Totaro, and M. Bayoumi, “Generative adversarial networks in security: a survey,” in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, Oct. 2020, pp. 0399–0405, doi: 10.1109/UEMCON51285.2020.9298135.

[9] L. Zhang, Z. Wang, and B. Wang, “Improving differential-neural cryptanalysis,” *Cryptology ePrint Archive*, 2022.

[10] B. Zolfaghari and T. Koshiba, “AI makes Crypto evolve,” *Applied System Innovation*, vol. 5, no. 4, Jul. 2022, doi: 10.3390/asi5040075.

[11] F. Barbosa, A. Vidal, and F. Mello, “Machine learning for cryptographic algorithm identification,” *Journal of Information Security and Cryptography (Enigma)*, vol. 3, no. 1, pp. 3–8, Sep. 2016, doi: 10.17648/enig.v3i1.55.

[12] M. S. Naik, D. K. Sreekantha, and K. V. S. S. S. S. Sairam, “A comparative study of the implementation of block ciphers for devices with limited resources (review),” (in Ukranian) *News of higher educational institutions. Radio electronics*, vol. 66, no. 3,

pp. 148–163, doi: 10.20535/s0021347023050011.

[13] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, “GIFT: a small present,” in *Cryptographic Hardware and Embedded Systems – CHES 2017. CHES 2017. Lecture Notes in Computer Science*, vol. 10529, W. Fischer and N. Homma, Eds. Springer Cham, 2017, pp. 321–345.

[14] D. Dinu, L. Perrin, A. Udovenko, V. Velichkov, J. Großschädl, and A. Biryukov, “SPARX: A family of ARX-based lightweight block ciphers provably secure against linear and differential attacks,” in *NIST Lightweight Cryptography Workshop 2016*, 2016, pp. 44–75.

[15] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, “RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms,” *Science China Information Sciences*, vol. 58, no. 12, pp. 1–15, Nov. 2015, doi: 10.1007/s11432-015-5459-7.

[16] J. Borghoff *et al.*, “PRINCE – a low-latency block cipher for pervasive computing applications,” in *ASIACRYPT 2012: Advances in Cryptology – ASIACRYPT 2012 – Lecture Notes in Computer Science*, Vol. 7658., X. Wang and K. Sako, Eds. Berlin, Heidelberg: Springer, 2012, pp. 208–225.

[17] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, “The LED block cipher,” in *CHES 2011: Cryptographic Hardware and Embedded Systems – CHES 2011 – Lecture Notes in Computer Science*, vol. 6917, B. Preneel and T. Takagi, Eds. Berlin, Heidelberg: Springer, 2011, pp. 326–341.

[18] R. AlTawy and M. Youssef, “CryptoGAN: generative adversarial networks for cryptographic algorithm generation,” in *2020 IEEE Symposium on Security and Privacy (SP)*, May 2020, pp. 1594–1608, doi: 10.1109/SP48185.2020.00160.

[19] Z. Bao, J. Guo, M. Liu, L. Ma, and Y. Tu, “Enhancing differential-neural cryptanalysis,” in *ASIACRYPT 2022: Advances in Cryptology – ASIACRYPT 2022 – Lecture Notes in Computer Science*, Vol 13791., S. Agrawal and D. Lin, Eds. Springer, Cham, 2022, pp. 318–347.

[20] C. Brunetta and P. Picazo-Sanchez, “Modelling cryptographic distinguishers using machine learning,” *Journal of Cryptographic Engineering*, vol. 12, no. 2, pp. 123–135, Jun. 2022, doi: 10.1007/s13389-021-00262-x.

[21] G. Hill and X. Bellekens, “CryptoKnight: generating and modelling compiled cryptographic primitives,” *Information*, vol. 9, no. 9, p. 231, Sep. 2018, doi: 10.3390/info9090231.

[22] M. Rana, Q. Mamun, and R. Islam, “Enhancing IoT security: an innovative key management system for lightweight block ciphers,” *Sensors*, vol. 23, no. 18, pp. 1–25, Sep. 2023, doi: 10.3390/s23187678.

[23] G. Kim, H. Kim, Y. Heo, Y. Jeon, and J. Kim, “Generating cryptographic s-boxes using the reinforcement learning,” *IEEE Access*, vol. 9, pp. 83092–83104, 2021, doi: 10.1109/ACCESS.2021.3085861.

[24] R. Hamza and D. Minh-Son, “Privacy-preserving deep learning techniques for wearable sensor-based big data applications,” *Virtual Reality & Intelligent Hardware*, vol. 4, no. 3, pp. 210–222, Jun. 2022, doi: 10.1016/j.vrih.2022.01.007.

[25] I. Negabi, S. Ait El Asri, S. El Adib, and N. Raissouni, “Convolutional neural network based key generation for security of data through encryption with advanced encryption standard,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 2589–2599, Jun. 2023, doi: 10.11591/ijece.v13i3.pp2589-2599.

[26] S. Bharati and P. Podder, “Machine and deep learning for IoT security and privacy: applications, challenges, and future directions,” *Security and Communication Networks*, vol. 2022, pp. 1–41, Aug. 2022, doi: 10.1155/2022/8951961.

[27] A. Abdulgadir, S. Lin, F. Farahmand, J.-P. Kaps, and K. Gaj, “Side-channel resistant implementations of a novel lightweight authenticated cipher with application to hardware security,” in *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, Jun. 2021, pp. 229–234, doi: 10.1145/3453688.3461761.

[28] N. Ren, D.-Y. Lv, J.-B. Wang, and X.-Y. Wang, “Solution algorithms for single-machine scheduling with learning effects and exponential past-sequence-dependent delivery times,” *Journal of Industrial and Management Optimization*, vol. 19, no. 11, pp. 8429–8450, 2023, doi: 10.3934/jimo.2023045.

[29] A. Anees, I. Hussain, U. M. Khokhar, F. Ahmed, and S. Shaukat, “Machine learning and applied cryptography,” *Security and Communication Networks*, vol. 2022, pp. 1–3, Jan. 2022, doi: 10.1155/2022/9797604.

[30] M. M. Alani, “Applications of machine learning in cryptography: A survey,” in *ACM International Conference Proceeding Series*, Jan. 2019, pp. 23–27, doi: 10.1145/3309074.3309092.

[31] A. Gohr, “Improving attacks on round-reduced Speck32/64 using deep learning,” in *Advances in Cryptology – CRYPTO 2019 – Lecture Notes in Computer Science book series*, vol. 116., A. Boldyreva and D. Micciancio, Eds. Springer, Cham, 2019, pp. 150–179.

[32] A. Benamira, D. Gerault, T. Peyrin, and Q. Q. Tan, “A deeper look at machine learning-based cryptanalysis,” in *Advances in Cryptology – EUROCRYPT 2021 – Lecture Notes in Computer Science*, vol. 12696 LNCS, A. Canteaut and F.-X. Standaert, Eds. Springer International Publishing, 2021, pp. 805–835.

[33] H. Kim *et al.*, “Deep-learning-based cryptanalysis of lightweight block ciphers revisited,” *Entropy*, vol. 25, no. 7, pp. 1–14, Jun. 2023, doi: 10.3390/e25070986.

[34] J. So, “Deep learning-based cryptanalysis of lightweight block ciphers,” *Security and Communication Networks*, vol. 2020, pp. 1–11, Jul. 2020, doi: 10.1155/2020/3701067.

[35] E. Biham, O. Dunkelman, and N. Keller, “New combined attacks on block ciphers,” in *Fast Software Encryption - Lecture Notes in Computer Science*, vol. 3557, H. Gilbert and H. Handschuh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 126–144.

[36] P. Sasikumar, “High-order DPA analysis of speck cipher with masking countermeasures,” *IEEE TIFS*, vol. 68, no. 3, pp. 2208–2220, 2023, doi: 10.1109/TIFS.2022.3195644.

[37] A. Ibrahim, H. Nemati, T. Schlüter, N. O. Tippenhauer, and C. Rossow, “Microarchitectural leakage templates and their application to cache-based side channels,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, Nov. 2022, pp. 1489–1503, doi: 10.1145/3548606.3560613.

[38] K. B. Jithendra and S. T. Kassim, “New results in biclique cryptanalysis of full round GIFT,” *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 5, pp. 5551–5560, Nov. 2021, doi: 10.3233/JIFS-189875.

[39] H. Arabnezhad-Khanoki and B. Sadeghiyan, “Toward a more efficient Grobner-based algebraic cryptanalysis,” *Journal of Computing and Security*, vol. 7, no. 2, pp. 103–117, 2020, doi: 10.22108/jcs.2020.123673.1050.

[40] R. Anand, A. Maitra, S. Maitra, C. S. Mukherjee, and S. Mukhopadhyay, “Quantum Resource Estimation for FSR Based Symmetric Ciphers and Related Grover’s Attacks,” in *Progress in Cryptology – INDOCRYPT 2021 – Lecture Notes in Computer Science*, A. Adhikari, R. Küsters, and B. Preneel, Eds. Springer, Cham, 2021, pp. 179–198.

[41] C. Zhao *et al.*, “Secure multi-party computation: theory, practice and applications,” *Information Sciences*, vol. 476, pp. 357–372, Feb. 2019, doi: 10.1016/j.ins.2018.10.024.

[42] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine

learning," in *ASIA CCS 2017 - Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security (2017)*, 2017, pp. 506–519, doi: https://dx.doi.org/10.1145/3052973.3053009.

[43] M. Favaretto, E. De Clercq, and B. S. Elger, "Big data and discrimination: perils, promises and solutions. A systematic review," *Journal of Big Data*, vol. 6, no. 1, pp. 1–27, Dec. 2019, doi: 10.1186/s40537-019-0177-4.

[44] J. Gilmer, V. Vig, and B. Shashua, "Explainable AI: A seven-point plan," *arXiv preprint arXiv:1705.07894*, 2017.

[45] R. Zhang, S. Hussain, H. Chen, M. Javaheripi, and F. Koushanfar, "Systemization of knowledge: robust deep learning using hardware-software co-design in centralized and federated settings," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 6, pp. 1–32, Nov. 2023, doi: 10.1145/3616868.

[46] K. Jang, G. Song, H. Kim, H. Kwon, H. Kim, and H. Seo, "Efficient implementation of PRESENT and GIFT on quantum computers," *Applied Sciences*, vol. 11, no. 11, pp. 1–16, May 2021, doi: 10.3390/app11114776.

[47] J. -F. Biasse, X. Bonnetain, E. Kirshanova, A. Schrottenloher, and F. Song, "Quantum algorithms for attacking hardness assumptions in classical and post-quantum cryptography," *IET Information Security*, vol. 17, no. 2, pp. 171–209, Mar. 2023, doi: 10.1049/ise2.12081.

[48] D. D. Tran, K. Ogata, S. Escobar, S. Akleylek, and A. Otmani, "Kyber, saber, and SK-MLWR lattice-based key encapsulation mechanisms model checking with maude," *IET Information Security*, vol. 2023, pp. 1–17, Oct. 2023, doi: 10.1049/2023/9399887.

[49] J. Zhang *et al.*, "Implementation of classic McEliece key generation based on Goppa binary code," in *2022 IEEE 16th International Conference on Solid-State & Integrated Circuit Technology (ICSICT)*, Oct. 2022, pp. 1–3, doi: 10.1109/ICSICT55466.2022.9963372.

[50] M. Shaikh, Q. A. Arain, and S. Saddar, "Paradigm shift of machine learning to deep learning in side channel attacks - a survey," in *2021 6th International Multi-Topic ICT Conference (IMTIC)*, Nov. 2021, pp. 1–6, doi: 10.1109/IMTIC53841.2021.9719689.

[51] M. Wazid, A. K. Das, V. Chamola, and Y. Park, "Uniting cyber security and machine learning: advantages, challenges and future research," *ICT Express*, vol. 8, no. 3, pp. 313–321, Sep. 2022, doi: 10.1016/j.icte.2022.04.007.

[52] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina, "SoK: deep learning-based physical side-channel analysis," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–35, Nov. 2023, doi: 10.1145/3569577.

BIOGRAPHIES OF AUTHORS

Mahendra Shridhar Naik received the B.E. degree in electronics and communication engineering from Visvesvarayya Technological University, Belagavi, in 2009 and the M. Tech. degrees in digital electronics and communication from Visvesvarayya Technological University, Belagavi in 2013. Currently, he is an assistant professor at the Department of Electronics and Communication Engineering, New Horizon College of Engineering, Bengaluru, India. His research interests include machine learning, deep learning, cryptography, networking and artificial intelligence. He can be contacted at email: mahendrasnaik@gmail.com.

Madhavi Mallam holds a PhD in signal processing from JNTU, Kakinada. She is currently working as Professor and Head at PES Institute of Technology and Management, Shivamogga, India. Her area of interests includes signal processing, image processing, speech processing, VLSI and machine learning. She can be contacted at email: gurumadhu432@gmail.com.

Chaitra Soppinhalli Nataraju received the B.E. degree in electronics and communication engineering from Visvesvarayya Technological University, Belagavi, in 2010 and the M.Tech. degrees in digital electronics and communication from Visvesvarayya Technological University, Belagavi in 2013. Currently, she is an assistant professor at the Department of Electronics and Communication Engineering, GM Institute of Technology, Davangere, India. Her research interests include VLSI, body area networks, communication, networking and machine learning. He can be contacted at email: chaitrasn48@gmail.com.