Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Dynamic Key Generation With LSTM: A Robust Defence Against DPA Attacks on FPGA-based AES Cryptosystems

Khalil M. Abdelnaby

k.abdelnaby@ammanu.edu.jo

Al-Ahliyya Amman University

Mohammed A. F. Al-Husainy

Al-Ahliyya Amman University

Mohammad O. Alhawarat

Al-Ahliyya Amman University

Mohamed A. Rohaim

Al-Ahliyya Amman University

Khairy M. Assar

Al Azhar University

Article

Keywords: FPGA, Smart grids, Dynamic Key generation, AES data encryption, LSTM network, Cryptographic Accelerator, Side-Channel Attacks, DPA attacks, Plain text, Cipher text

Posted Date: March 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9006459/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

DYNAMIC KEY GENERATION WITH LSTM: A ROBUST DEFENCE AGAINST DPA ATTACKS ON FPGA-BASED AES CRYPTOSYSTEMS

Khalil M. Abdelnaby[1,2]. Mohammed A. F. Al-Husainy[1]. Mohammad O. Alhawarat[1]. Mohamed A. Rohim[1,2]. Khairy M. Assar[2]
[1]Faculty of Information Technology, Al-Ahliyya Amman University, Amman 19328, Jordan
[2]Systems and Computers Engineering Department, Faculty of Engineering, Al-Azhar University, Nasr City 11765, Cairo, Egypt.
Corresponding Author: **Khalil M. Abdelnaby** (k.abdelnaby@ammanu.edu.jo)

Abstract: This research presents a new, hardware-efficient cryptosystem that integrates a Long Short-Term Memory (LSTM) network with the Advanced Encryption Standard (AES) on an FPGA to create a strong first line of defense against Differential Power Analysis (DPA) attacks. The key innovation is a dynamic key generation scheme that uses an on-chip NIOS II processor to run a pre-trained LSTM model to create non-deterministic, session-specific round keys in real-time, breaking the statistical relationships that side-channel attackers exploit to create their attacks. The system is implemented in a Cyclone IV DE2-115 FPGA, and was thoroughly tested using functional simulation and physical synthesis. The architecture has an impressive throughput of 989.94 Mbps, at a maximum clock speed of 400.198 MHz, indicating that it is possible to have a high level of security without sacrificing performance. The comprehensive side-channel analysis, including correlation power analysis and Test Vector Leakage Assessment (TVLA) t-tests, has validated that there is nothing that can be used to create DPA attacks, showing that there was no usable leakage of data after >100,000 power traces were recorded. The consumption of 12.5% of slice registers and 14.5% of LUTs, although a burden for the design to include the neural key scheduler and soft-core processor, is justified, and the dynamic power consumption of the design is only 0.5 mW. Thus, this work provides a new direction for designing DPA-resistant cryptographic accelerators, proving that the use of machine learning-generated dynamic keys can significantly increase the security of these systems.

Keywords FPGA. Smart grids. Dynamic Key generation. AES data encryption. LSTM network. Cryptographic Accelerator. Side-Channel Attacks. DPA attacks. Plain text. Cipher text.

I. Introduction

The future of electric power distribution is based on Smart Grid Technology. Smart grids use automatic controls, advanced monitoring, and demand-response load management to provide a more efficient, reliable, and secure method of electric power distribution [1, 2]. Smart grids are made up of nodes, communications systems, primary nodes (Substations, generation locations), and consumer energy appliances [3, 4]. Smart grids also allow for two-way communication, which allows electricity to be provided from suppliers to consumers based upon their respective demands [5, 6].

As is depicted in the diagram in **Fig. 1**, smart grids have a centralised architecture and rely on the Internet, making them susceptible to a variety of cybersecurity threats [7, 8]. The following steps must be taken to preserve cybersecurity for smart grids: (1) Sensitive data must be kept confidential and only released to individuals who are duly authorized to have access to sensitive data; (2) The integrity of sensitive data must not be damaged or modified by any unauthorized parties; and, (3) Access to sensitive data must be made available to duly authorized individuals at their request. Cyberattacks can be mitigated using cryptographic means; however, hardware-based encryption modules provide more robust security and flexibility than other methods [3, 9, 10].

A Field Programmable Gate Array (FPGA) is a computing device that provides the advantages of an Application-Specific Integrated Circuit (ASIC), specifically, its ability to perform many tasks in parallel at high speed while still maintaining the ability to be reprogrammed using hardware description languages (HDLs). An FPGA can be designed to support various hardware protocols. Compared to ASICs, FGPA allows for easy modification, high performance, and customization for specific applications. Compared to ASICs, FPGAs provide significantly higher design flexibility, rapid reconfigurability, and lower non-recurring engineering costs, making them particularly suitable for prototyping and low-to-medium production volumes. While ASICs generally achieve higher performance and lower power consumption due to full-custom optimization, FPGAs offer a practical trade-off between efficiency and adaptability for security-critical and evolving applications [11, 12].

The adaptability and versatility of FPGAs can be noticed in the different fields using the particular chip, which includes but is not limited to the telecommunications industries, the computer industries, the industrial control industries, and their manufacturing processes for Medical Devices. Considering data security, FPGAs are employed in building encryption modules to guard sensitive data in smart grids, Supervisory Control and Data Acquisition (SCADA) systems, and Defense Networks. They represent a practical and widely adopted platform for securing critical data,

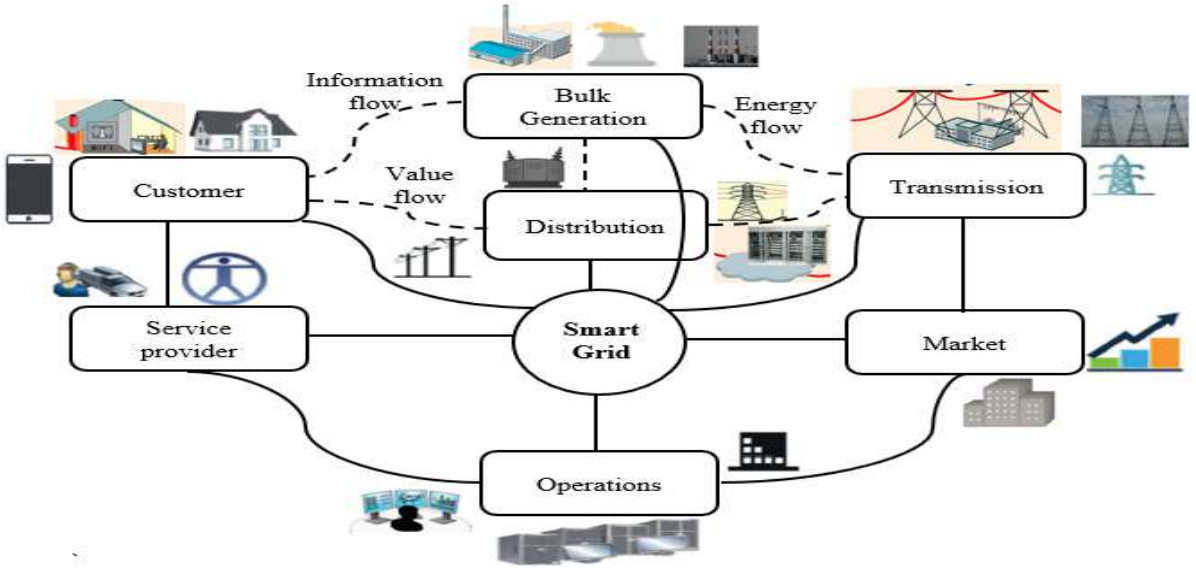


Fig. 1. A general model of smart grid communication network architecture.

particularly in scenarios that require post-deployment reconfiguration, rapid prototyping, or evolving cryptographic requirements [3, 13].

Deep Learning is a machine learning technique involving artificial neurons in the learning process. Various tasks have utilized Deep Learning, such as Image Classification, Natural Language Processing (NLP), and Speech Recognition [14-20]. Deep Learning can be applied to various security-related tasks in the Electricity Sector, such as Fraud Analysis, Intrusion Detection, Malware Identification, as well as Asset Protection [21, 22].

LSTM (Long Short-Term Memory) is a Deep Learning Network architecture that imitates the ability of the biological human neural system. Examples of LSTM applications include Data Classification, Object Detection, Data Encryption, Decryption, and Key Generation. LSTM Networks are composed of two main parts, which include layers LSTM and Fully Connected Layers [23]. The purpose of the LSTM layers is to ensure that the network can store a series of data, and that of the Fully Connected Layers is to optimize the parameters by the use of non-linear functions [24]. AES (Advanced Encryption Standard) is a highly secured Encryption Algorithm.

This paper will introduce a new LSTM (Long Short-Term Memory) architecture that will enhance the production of AES encryption keys used in Smart Cryptosystems as well as securing it against possible threats. The new LSTM encryption/decryption architecture will enable the implementation of AES encryption algorithms to be an easier method than other AES methods that normally do not utilize non-linear functions. Moreover, this architecture enables reversibility of encryption, besides FPGA compatibility, to a lesser error rate compared to traditional approaches.

The research provides an architectural description and a software modification of the LSTM architecture, and provides simulation results of an implementation of the AES-128 functional block with its internal LSTM components. Also, the Authors highlight the need for further research on some of the implementation issues regarding support for FPGAs, such as the implementation of a 128-bit AES-based LSTM on the Cyclone IV Altera DE2-115-based Solid-State Drive (SSD), including descriptions of the specific hardware used, timing issues related to this platform, and the security impacts related to Smart Grids. The HDL code generated has been synthesized, implemented and validated using test benches for the development of this new architecture and the Authors report that they have achieved significant improvements in several areas including Chip Efficiency, The Speed and Performance of Chips, The Size of Future Chips, Their Power Consumption as well as Their Overall Security, using FPGA-based platforms for the purpose of demonstrating their LSTM algorithm is ready to be deployed on a production ready basis for use as a 128-bit encrypted/decrypted algorithm for secure/semi-secure Critical Infrastructure and to enhance both the overall security and operational safety of Critical Infrastructure.

The following contributions have been made to this area of research:

(i) A novel LSTM-based rolling key generation mechanism is proposed to enhance the security and performance of AES encryption.

(ii) The LSTM architecture is optimized for FPGA implementation, enabling efficient and reliable integration with AES encryption and decryption processes.

(iii) A comparative evaluation is conducted against existing FPGA-based encryption systems, demonstrating

improvements in data throughput, maximum operating frequency, and hardware efficiency.

(iv) The system is successfully implemented on a Cyclone IV FPGA platform, with detailed resource-optimization strategies to address FPGA hardware constraints.

(v) Compared to prior LSTM-based implementations, the proposed architecture achieves reduced energy consumption and lower architectural complexity.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents the problem formulation and the proposed architecture. Section IV describes the experimental setup and evaluation methodology. Section V discusses the experimental results. Finally, Section VI concludes the paper and summarizes the main findings.

II. Related Work

Cyber-physical infrastructures such as smart grids, industrial control systems, and distributed sensor networks are increasingly exposed to cyber threats that directly affect operational reliability and safety. The smart-grid environment, especially, is a heterogeneous wired and wireless communication environment, which collects sensor states, power-line measurements, and control signals representing a significant enlargement of the attack surface and creates complex authentication and authorization challenges [25, 26]. The user authentication, device identification, and secure communication protocols have thus become key components of contemporary grid security arrangements [25, 27]. Iyer *et al.* [28] also highlight that with the increased size of grid infrastructures, the number of network nodes correspondingly increases to expose vulnerabilities, which means that secure cryptography, and with an effective key-management solution, are all required to enable large infrastructures without compromising performance.

Regarding the hardware implementation, much effort has been put into optimizing the Advanced Encryption Standard (AES) for embedded and FPGA systems to strike a balance between throughput, silicon area, and power consumption. Rao *et al.* [29] suggested an architecture of AES that is memory efficient, suggesting that round-key scheduling can be optimized to achieve a low hardware footprint and high encryption rates. To enhance reliability, complementary to the study by Mestiri *et al.,* [30], modified temporal redundancy strategies to identify faults in multiple rounds of AES were investigated to enhance application in safety-critical domains. FPGA cryptographic design verification and validation pipelines were studied by Yang *et al.* [31], who showed RTL-level co-simulation techniques to verify that the design is functional at the RTL level before being run on hardware.

Similar research lines have covered hardware trust, FPGA security primitives, and supply-chain vulnerabilities. The survey of FPGA-based Physically Unclonable Functions (PUFs) by Anandakumar et al. [32] found both attack vectors and countermeasures, and the analysis of the security properties of ARM TrustZone in the devices, namely AMD-Xilinx Zynq-7000 SoC, by Benhani *et al.* [33]. The comparative evaluation of vendor-level control and configuration mechanisms in the key FPGA vendors was done in [34], and the risks of the supply chain to FPGA-based systems were reviewed by Zhang and Qu. Jin *et al.* [36] provided broader threat analyses in relation to the use of FPGA in cloud computing infrastructures and identified configuration leakage and multi-tenant attack surfaces. The article by Zheng *et al.* [37] concentrated on verification automation by assertion-based simulation techniques during the RTL level and the gate level, which supports the significance of systematic validation processes. Other application-related works like the FPGA-based solar-wind hybrid power management system by Nagaraj *et al.* [38] are yet another (more) demonstration of the use of FPGA technology in mission-critical and energy-sensitive applications.

A much smaller literature will focus on AES implementation in reconfigurable logic. Kouser *et al.* [39] showed a 128-bit AES design based on Spartan-6 and using iterative looping techniques to provide real-time encryption and use the reconfigurability of the FPGA. But most of these implementations are focused on performance or area reduction and not adaptive security features. A more recent focus has been on combining machine learning and neural computation with cryptographic operations to make them more adaptable and resilient. In [40] and [41], neural-assisted authentication and secure communication protocols of 5G and IoT ecosystems have been suggested that integrate symmetric encryption with intelligent network-selection schemes, which have been verified by formal logic tools like Burrow-Abadi-Needham analysis. Neural-based encryption methods have been considered to encrypt optical imagery [16], generate session keys in the IoT [17], image encryption by a model based on the concepts of fractional chaos [18], to multimedia cloud security [19], to encrypt medical data privately [20] are some examples of how artificial intelligence is finding increased use in the development of cryptography. The practicality of the use of convolutional neural networks in AES-based generation of key was discussed in [24], with the notion of cryptographic intelligence systems, a combination of pattern recognition and symmetric encryption.

The ability to deploy neural networks efficiently on FPGA platforms has been approached with a lot of effort on the hardware-acceleration front. F-LSTM proposed by Bushun *et al.* [42] is a heterogeneous architecture that allows the acceleration of long short-term memory algorithms on

reconfigurable logic, highlighting that the community has been more interested in CNNs than recurrent designs. Guy *et al.* [43] proposed a stochastic-computing-based LSTM inference engine utilizing less hardware and less power and maintains computational accuracy. The convolutional IP core based on the reuse of both temporal and spatial data on Zynq UltraScale+ systems was introduced by Nguyen *et al.* [44], and it was able to show high-throughput deep-learning inference acceleration. There were also hybrid neural accelerators described in [45] (Auto-Encoders and LSTMs) and dedicated FPGA LSTM-RNN accelerators described in [46] (90 GFLOPs). In the FPGA-based AI system, real-time CNN-based UAV surveillance systems were introduced in [47], highlighting the appropriateness of reconfigurable logic in AI applications with high latency requirements. The Tandem Deep Learning Side-Channel Attack presented in [48], on the other hand, demonstrates that neural algorithms can also be used to increase adversarial capabilities by reducing the quantity of traces needed to extract keys, which further justifies the need to make hardware-level defense improvements.

Although the current research environment is quite substantial, the majority of the earlier works either maximize the AES throughput and area metrics or externalize neural models to authenticate, detect anomalies, or classify data. Elimination of the static cryptographic secrets at the implementation level is a relatively unexplored field. Traditional countermeasures, including masking, hiding, and dynamic reconfiguration, and based on PUF, are the main approaches to safeguard a fixed secret key instead of eliminating the architectural reliance on persistent keys. To fill this gap, the current work proposes a solution that involves the integration of an LSTM-based round-key generation mechanism using an FPGA and integrating it into the AES pipe. The idea works to modify the defensive model of the static-key protection to the static-key elimination to achieve a higher resistance to Differentiating Power Analysis (DPA) and maintain high throughput, architectural reconfigurability, and an embedded deployment environment.

III. Problem Description and Proposed System Architecture

In light of increases in sophisticated threats to sensitive data stemming from exploiting Differential Power Analysis (DPA) weaknesses residing in the Advanced Encryption Standard (AES), the current cryptographic climate faces numerous challenges in addressing those threats. At present, all current Encryption Technologies are considered "secure" and continue to maintain a measure of confidentiality against criminal activity; however, because of the potential for today's continuing advancement of both technology and sophistication levels of crime, the above-mentioned methods of encryption, while again "secure," will no longer be fully sufficient in addressing such evolving threats. Another major factor in the use of traditional AES implementations is that they tend to require an inordinate number of resources to complete their work and thus, create a bottleneck in the ability to perform cryptographic operations securely and efficiently, particularly in FPGA environments [49, 50].

To combat the issues related to DPA attacks of AES and improve the ability to perform cryptographic operations, we propose to provide a cutting-edge approach to creating custom hardware for creating dynamic encryption keys in conjunction with AES. We will implement a detailed approach to the design and creation of an LSTM-based hardware architecture for creating the Encryption Keys required for AES-128 using smart crypto systems, based upon the knowledge of and input to the operation of this architecture.

The proposed architecture will ease the application of encryption with classical AES schemes and will be capable of being used on FPGAs that will provide a reversible architecture that will encrypt and decrypt plaintext/ciphertext with high efficiency and with minimal error rates. In this regard, we will give a detailed comparison of different FPGA implementations in terms of their major performance parameter,s such as Data Mode, Maximum Frequency, Throughput, Area Utilization, and Efficiency, as to why our proposed cryptographic system has further advantages over the existing ones.

With the implementation of our LSTM-based key generation scheme on a Cyclone IV FPGA, we shall demonstrate how FPGA resources can be utilized effectively by the architecture of this innovative system when using it to encryption via AES-128. Although the Cyclone IV FPGA does not have a considerable amount of physical memory resources, the suggested solution will prove that high efficiency improvement can be obtained in comparison to the currently existing AES-based approaches that require less power and offer an easy way of implementation. This novel algorithm, in addition to making DPA attacks of encryption techniques based on AES less likely, preconditions a more secure, energy-efficient, and scalable way of cryptographic applications using FPGAs.

A. Threat Model and Security Objectives

A sound and justifiable security analysis requires the clear and formal description of the adversarial environment, the assumptions about the system, and the very specification of the scope of the intended protection. In the absence of this formalization, claims of resistance or robustness are prone to becoming ambiguous, misconstrued, or exaggerated, as well. In turn, this paper uses a carefully scoped and practically inspired threat model that is carefully adjusted to the current body of knowledge on side-channel and embedded cryptographic attacks. This model is the guiding paradigm

on which all the following claims and experimental findings concerning security are strictly assessed [51].

- **Adversarial Capabilities and Operational Boundaries**: The adversary in this model has been assumed to have non-invasive physical access to the target cryptographic device and is capable of measuring visible external side-channel emanations, where the key object of interest is the dynamics of power consumption. This attacker can obtain a sufficiently large number of high-fidelity traces of power when active encryption is being performed, and he is authorized to use third-rate statistical aftermath strategies, such as Correlation Power Analysis (CPA), Differential Power Analysis (DPA), and advanced higher-order statistical distinguishers. There are, however, strict limits placed on adversarial capability: the attacker is not presumed to be able to access internal FPGA routing or configuration memory, maliciously modify bitstreams or inject specific faults, maliciously perform invasive micro probing or laser fault injection, or control external environmental parameters such as supply voltage or temperature outside of nominal operating conditions. This construct also uniquely represents a powerful yet realistic laboratory or field attacker, in contrast to an unbounded resources adversary with a nation-state opponent and the use of invasive hardware methods.
- **Observable Phenomena and the Defined Attack Surface**: In this bounded model, the observable that is the most visible is the instantaneous and cumulative power consumption of the time-series at the supply line of the device. Theoretically, electromagnetic emissions have the same content of information, but they are not measured quantitatively in the current research. Timing information is not viewed as secret, though this is publicly visible. Most importantly, and in complete compliance with the principle stated by Kerckhoffs, all the parameters of the algorithm, such as the AES cipher structure, are supposed to be in the full public knowledge. The safety of the system is not based on an obscurity of algorithms, but rather on the safety of the dynamic, session-specific key material.
- **Foundational Security Assumptions**: There are several fundamental assumptions, based on which the integrity of the system operates. The most important is that there is a securely shared master secret key between communicating endpoints, and a trusted provisioning stage has been conducted even before operational deployment. This master secret is never passed across the dynamic key generation process and will never be stored in plaintext in external memory. Moreover, the communication medium in which the public synchronization parameters (session counters or nonces) are exchanged is assumed to be authentic (tamper-evident) but not necessarily confidential.
- **Primary and Secondary Security Objectives**: The ultimate security goal is to provide statistical key non-recoverability when the attack is in terms of trace accumulation. The architectural design, in particular, is meant to foil the capability of an adversary to effectively match the pattern of intermediate computational leakage throughout a series of encryption transactions, to deduce any underlying secret material. The architecture does not purport absolute, physical removal of instantaneous information leakage, a quest that is usually accompanied by impractical overheads. Rather, it is tactical in its attempt to break the cross-trace statistical alignment, raise the effective key entropy per session, and cause the power trace domain to be gone, and thus dramatically raise the cost of the computation and measurement effort to pass a successful key recovery attack. Secondary, but important, are the goals of ensuring perfect deterministic encryption-decryption accuracy, mathematical integrity of the underlying AES cipher, and reduction of hardware overhead in order to be deployed in the resource-constrained FPGA world. This clearly delimited threat model makes sure that all the further evaluation measures, experimental approaches, and security assertions are technically accurate, pragmatistically based, and formally interpretable.

B. AES-128 Encryption and Decryption Algorithm

AES-128 is the cryptographic engine for the proposed system. A new key management solution based on machine learning is proposed to address the modern-day cryptographic issues, especially DPA (Differential Power Analysis). This section describes how this hybrid approach uses a machine learning-based key generator to create keys for the AES encryption process.

As shown in **Fig. 2**, the system includes a Rolling Key Generator that feeds a dual-path Encryption/Decryption Engine that consists of both an AES encryptor and an AES decryptor with an LSTM-Based Crypto Key Expansion module (aka Round Key Generator). The main advantage of this dual-path approach is that it allows for continuous and secure communications between encrypting and decrypting devices. In this way, the system can take in Plain Text/Raw Data and convert it into Encrypted Data during transmission and vice versa at the time of receiving.

To ensure confidentiality, the Encryption Process (EP) uses an intricate combination of dynamic keys that enhance confidentiality and protect information:

- **Dynamic Key Generation via LSTM**: The keying material is generated using a Rolling Key Generator instead of having a fixed initial key. The Rolling Key Generator makes the seeds that the LSTM Network uses to come up with the encryption keys, which take advantage of its capability to model the complex time series and dependencies. Moreover, non-linear, context-sensitive, and prediction-resistant keys introduce both randomness and dynamism, not present in conventional AES Key Schedules, and

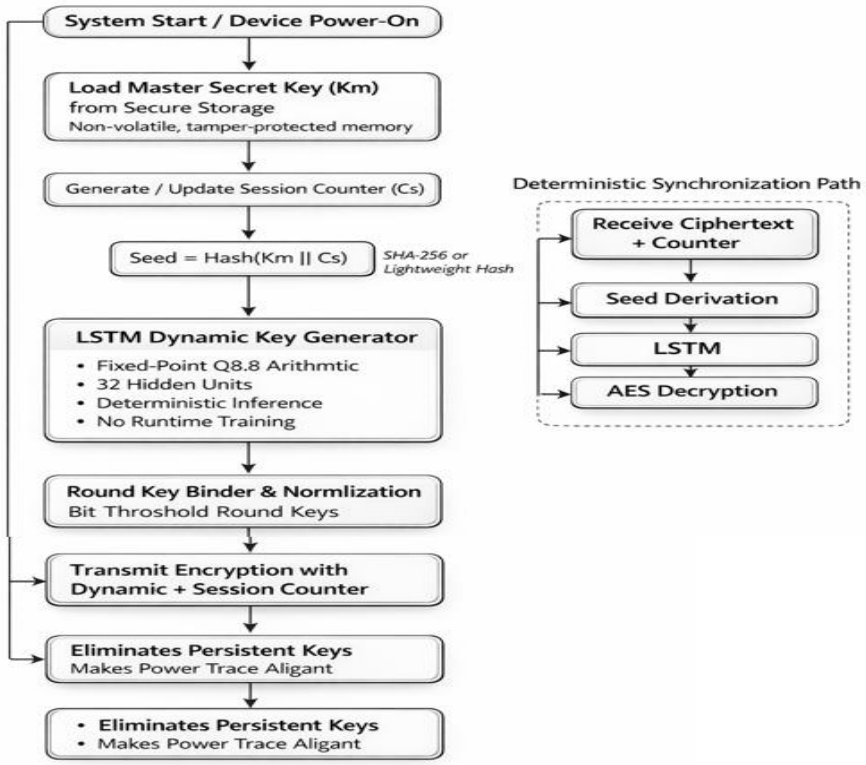


Fig. 2. The Schematic representation of AES algorithm for encryption and decryption round.

therefore they assist in an extra measure of protection against statistical analysis (e.g., DPA).

Intelligent Round Key Expansion: What the LSTM is generating as the primary cipher key is an additional Round Key Expansion of the initial key. Although this expansion is informed or perturbed by the internal state of the LSTM, and is based on the standard AES key schedule, this allows the derived round keys to be not a mere deterministic expansion of a known input. This smart growth forms the cryptographic strength of every round key, developing an even more intricate connection among them and additionally obscuring the connection between power burn and key bits.

- **Confidentiality Transformation with AES-128**: AES Encryptor encrypts plaintext using AES that enables a series of rounds (ten rounds with AES-128) in a series of operations; SubBytes, ShiftRows, MixColumns, and AddRoundKey. The only significant change is the fact that round keys derived off an LSTM-based system are used in the AddRoundKey operation of each round. The AES process contains cryptographic properties that ensure confidentiality and integrity of the encrypted data and is used to prevent attacks that might have tried to decrypt the encrypted data by using transformed but mathematically correct round keys.
- **Secure Data Emission:** The generated Ciphertext of the system is to be intentionally decrypted using Secure, Encrypted, or authentication through Secure-Socio Technical means. This solution is secured using the virtue of using AES but has been strengthened by the application of another method of generating keys, which is known as Dynamic Keying and this provides further protection against eavesdropping and key cryptanalysis.

Decryption is the symmetric inverse of encryption, meticulously designed to recover the original data accurately.

- **Synchronized Key Reconstruction**: The system can use the same order of keys that was used to encrypt the information to reconstruct the key used first before decryption of the ciphertext. Parameters and a common initial state (or side channel) are used in the Rolling Key Generator and LSTM-Based Key Expansion feature to recover the same primary cipher key and all the following round keys. Reconstruction of such required keys is highly important because a minute deviation would not permit a proper decryption.
- **Inverse Round Key Application**: The AES Decryptor decrypts using an inverse cipher. The LSTM-generated round-key (that was made using LSTM) expands backward. To complete decryption, every algorithm uses the exact reverse of what was done in encryption: Inverse ShiftRows to the left, Inverse SubBytes, Inverse MixColumns, and also AddRoundKey, but the order of the keys is inverse. The decryption process consists of reversing the steps that were done to encrypt the ciphertext by systematically undoing all of the transformations used to encrypt it.

- **Data Recovery via AES-128 Decryption**: To produce the output of the raw plaintext or data, which was encrypted, the ciphertext is processed through the rounds of decryption. The usability of the AES algorithm guarantees that when you have the matched keys for decryption during this process, the output will be the original raw data or plaintext, completely restored as it was before encryption, to be identical to the input before encryption.
- **Implicit** Integrity **Verification**: The success of decryption is one of the main methods that are used to determine if the ciphertext has changed during transfer or if the decryption key was entered incorrectly, in which case, the resulting decoded information would appear nonsensical. Although the AES decryption process will usually be coupled with an independent means of verifying the integrity of the data, such as a MAC, the fact that the decryption process produces the correct output indicates that the data itself is relatively valid.

C. Security Synthesis and Contribution

The key generation and expansion processes of AES-128 utilize an LSTM, which improves upon the weakness of static-key cryptographic systems. The introduction of computational randomness and 'temporal moment' (an example of which would be the time that a computer takes to complete a certain task) means that there is no direct relationship between the primary keys, round keys, and the power signature of the algorithm used. Therefore, the mathematical soundness of AES would be unchanged by the introduction of this new design; rather, the implementation layer is made stronger against certain types of side-channel attacks, such as Differential Power Analysis, which correlate the power signature of the device running AES with key-dependent computations.

Using this new approach creates an effective and strong cryptographic system. It is a great solution for smart grids, industrial IoT types of devices, and communications systems where devices may be vulnerable and the data they handle is of a high level of sensitivity, as it combines the durability of traditional cryptography with the flexibility of learning or adaptive-based security.

D. LSTM Architecture and Training Configuration

The very security posture of the suggested cryptosystem is designed not with the solitary strength of the classical primitives, however a harmonious blend with a neural stochastic key diversification engine. It is a mechanism that makes use of a Long Short-Term Memory (LSTM) network in an unconventional, novel application: as a deterministic, non-linear entropy amplifier. Its objective is clearly unlike those of traditional semantic learning or predictive pattern management work, but is rather a controlled pseudo-random sequence transformer. The essence of this is to create session-specific cryptographic round keys that are maximally-entropic, time-varying, and statistically-unpredictable, and meet strict conditions of computational feasibility in embedded FPGA implementation. The method converts a fixed key schedule into a dynamic context-sensitive process [52, 53].

- **Architectural Design Considerations**: In contrast to the intensive models of the deep type that are characteristic of the vision or language fields, the installed LSTM is lightweight in design and resource-sensitive. The design principles governing it focus on deterministic reproducibility to give the algorithm perfect encryption-decryption synchronization, a small memory and logic implementation, constant and predictable inference latency, full compatibility with fixed-point arithmetic, and the absolute lack of stochastic runtime training. The architecture thus developed, as described in **Fig. 3**. The sigmoid function (s) is used as a gate activation, and the hyperbolic tangent (tanh) is used as a cell state. This central core is preceded by a 128-bit projection layer, which is fully connected, and end-to-end bitwise threshold mapping, which is used to normalize the output. This design injects adequate non-linearity, enabling entropy amplification and strictly enforcing hardware overhead prohibition.
- **Numerical Representation and Quantization**: Each numerical piece, namely, weights, biases, and activation, is quantized to a Q8.8 fixed-point representation to smoothly move to efficient FPGA implementation. This guarantees a tradeoff between critical numerical stability and hardware efficiency. As illustrated in the annotations of **Fig. 3**, the activation functions are performed through piecewise linear look-up tables in the on-chip memory blocks, eliminating floating-point units, and also leading to power-centric dynamic switching activity. More importantly, training is a completely offline task that comes before the hardware synthesis, which is optimized not to be predictive but to have cryptographic entropy properties. The training corpus is a pseudo-random binary sequence and structured perturbation streams, which are aimed at maximizing the output diversity and reducing the correlation. The important parameters one can choose are the Adam optimizer, 200 epochs, and a personalized loss function with entropy maximization and regularization, and early stopping on the convergence of Shannon entropy [53, 54].
- **Inference-Only Execution and Security Rationale:** As it has been emphatically pointed out in Fig. 3 the deployed system is used in a strict inference-only mode; there is no runtime learning on the FPGA. This ensures that it has deterministic latency, does not have power-trace variability with gradient updates and has no attack vectors that might

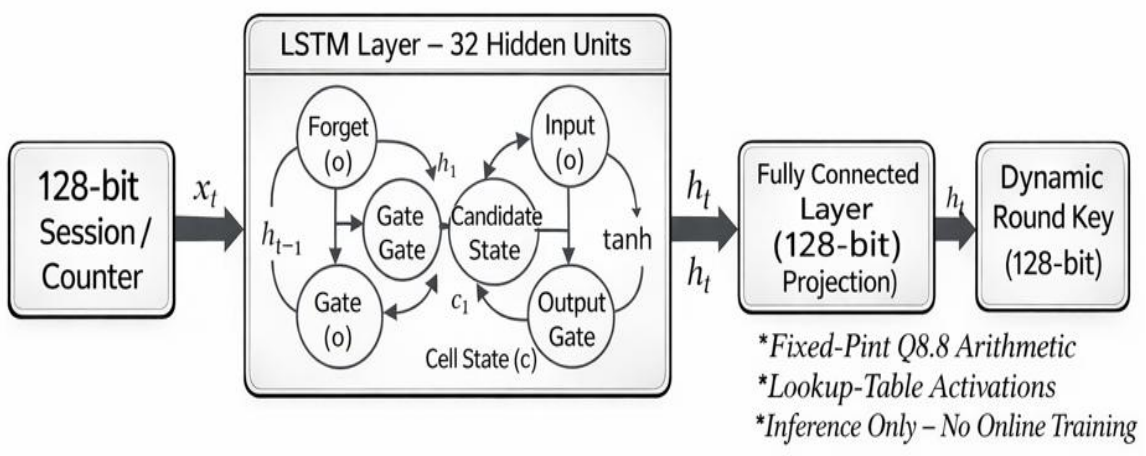


Fig. 3. Lightweight LSTM-Based Dynamic Key Generator Architecture

- enable adaptive behavior to be exploited. The security-motivated explanation of this neural part is multidimensional: it is not some analog of the existing AES key scheduling but a mathematically regulated perturbator. It is used to reject statistical alignment of power-trace, remove long-term material secret keys in memory, enhance effective keyspace diversity, and add temporal decorrelation between sessions. As a result, it can be seen that the system is a hybrid cryptographic-neural system that maintains the mathematical integrity of AES and essentially changes the implementation-side leakage profile of the system to make the traditional side-channel analysis far more complicated.

E. Dynamic Key Synchronization Protocol

The requirement of a strong, deterministic, and provable synchronization mechanism is the result of the introduction of a session-dependent dynamic round key regime. This protocol should ensure that end points of encryption and decryption generate the same sequences of keys without involving extensive mechanisms of exchange of keys, which would in turn render vulnerability. The suggested synchronization scheme can do so with a premise of deterministic seed derivation and neural inference symmetry, thus contributing zero extra communication expenses where the cryptographic integrity and operational trustworthiness are carefully upheld [55, 56].

- **Governing Design Principles and Protocol Philosophy:** The synchronization protocol is built based on principles that are consistently stable: Absolute Determinism, where each endpoint can produce round keys bit-identical to that of a single input; Minimal Communication, which forbids any exchange of keys between endpoints per round or per session; Replay Resistance, which is implemented by monotonically increasing identifiers of each session; Inherent Fault Tolerance, which ensures a temporary inconsistency in the view of a working end point; and Uncompromised Security Preservation, which denies that any secret material is leaked by synchronization metadata. All the above principles safe-guard that the protocol increases security without adding vectors of attack [56, 57].
- **Parameter Initialization and Public Input Framework**: Each communicating entity is initially provisioned with a shared Master Secret Key (K_m) in a secure, offline phase. During operational lifetime, both sides also maintain and observe a public Session Counter (C_s) or nonce. This counter is not secret; its purpose is solely to ensure temporal uniqueness and provide a monotonic, replay-resistant input to the key derivation function. The public nature of C_s is a deliberate design feature that eliminates any confidentiality requirement for the synchronization channel.
- **Deterministic Seed Derivation and Neural Symmetry**: For each cryptographic session, a deterministic 128-bit seed is derived via a cryptographic hash function: Seed = Hash ($K_m \parallel C_s$). This seed serves as the exclusive input vector to the identically configured LSTM key diversification network on both endpoints. Because both parties possess the identical K_m and observe the synchronized C_s value, they derive the same seed without any explicit communication of the seed itself. The LSTM network then executes a forward inference pass using this derived seed. Since the neural weights are fixed, static, and identical across both devices, the output—a 128-bit dynamic round-key sequence—is guaranteed to be deterministic and perfectly synchronized. This elegant symmetry transforms a shared secret and a public counter into a unique, session-specific key [58, 59].
- **Resilience Mechanisms and Security Rationale**: A lightweight resynchronization mechanism is called in case of a packet loss or counter drift; it only takes the transmission of the most recent, agreed counter value. Nothing in the form of cryptographic secrets is revealed in this recovery process. The protocol, therefore, gives security to forward secrecy of all the session keys and offers operational robustness. This is deeper in terms of security rationale, in that by eliminating persistent key material from the side-channel observable space in a regular manner and in a manner that makes each session produce statistically

independent round keys, the protocol cuts to the very root of cross-trace leakage accumulation, such as is the lifeblood of DPA and CPA attacks. Importantly, the synchronization mechanism does not add any cryptographic vulnerability area, since all of the exchanged parameters are not secret values, but just public indices to derive entropy.

F. Overview of the Proposed System Architecture

Fig. 4 depicts the advanced architecture of an FPGA-based ciphering system designed to protect information transmitted over the Internet by generating keys dynamically and using strong methods of ciphering. The design allows for the programmed processor to work with specialized hardware for encrypting and implementing diverse levels of protection in fast, flexible, and efficient modes of operation. Each element is detailed in this section, including what role it plays within the overall design.

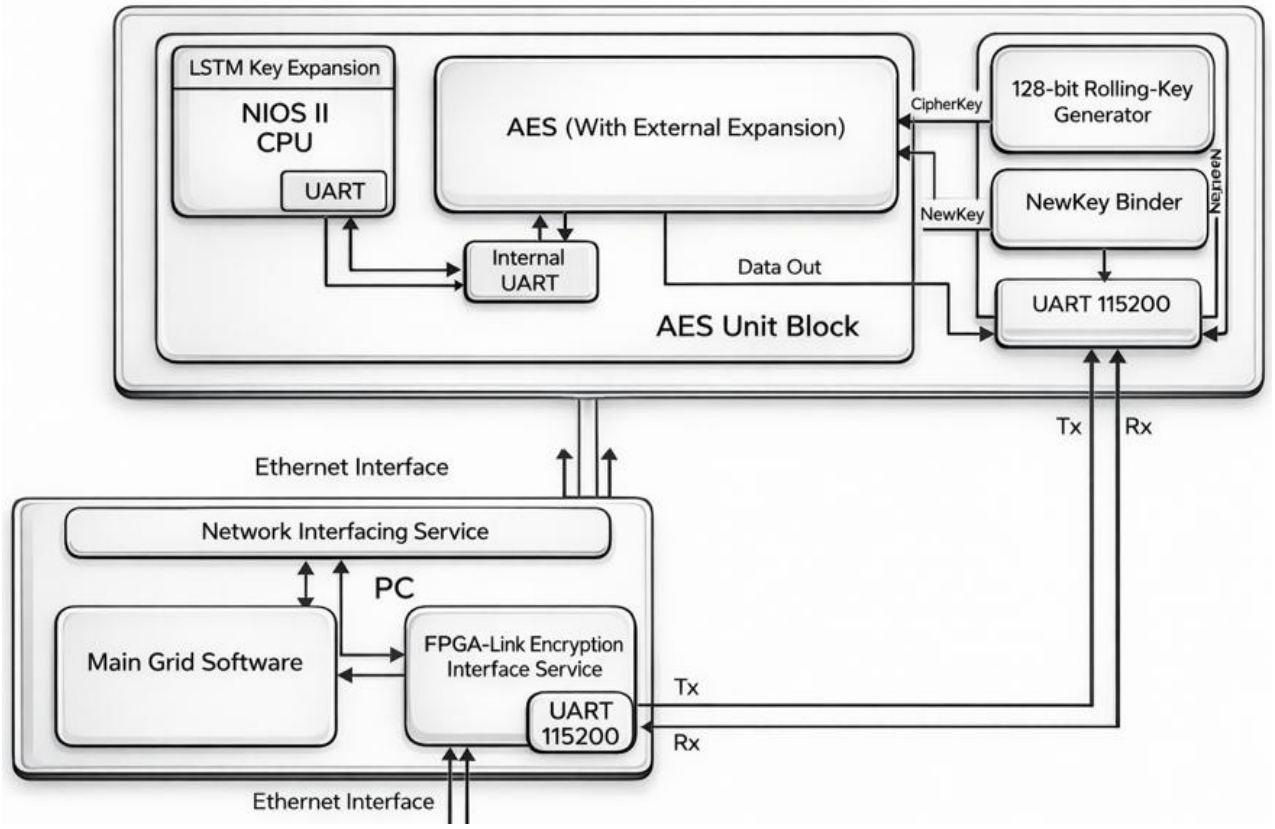


Fig. 4. The proposed system architecture

Central to the system is the AES Unit Block, which orchestrates the encryption and decryption processes. It incorporates several key sub-components that work in unison:

- **Soft-Core Altera NIOS II CPU:** As the brain of this system, the embedded CPU creates an LSTM-based Key Expansion Module capable of dynamically producing machine-learning-generated cryptographic keys using neural networks. An on-chip memory is used by the CPU to enable adaptive scheduling of input keys according to their current states, allowing for increased protection from several forms of brute force Cryptanalysis attack – including side-channel attacks, such as Differential Power Analysis (DPA), which rely on identifying patterns in a device's power consumption during the operation of a cryptographic algorithm.
- **Internal UART:** Inside the AES block, a Universal Asynchronous Receiver/Transmitter (UART) is integrated as an interface for communication internally (between devices) using a serial digital interface. This design creates a single, simple path for data to be exchanged between the NIOS II Microcontroller CPU and other system peripherals/modules and provides an efficient way to pass on control signals, encryption/decryption keys, and status data reliably.
- **AES with External Expansion:** To provide additional functionality to the primary AES Encryption/Decryption Engine, an external Key Expansion Logic module has been added. This module is powered by an LSTM Module that resides within the Central Processing Unit (CPU). As a result, the AES Engine can generate Dynamic Key Sequences between sessions and throughout the course of each session, thereby enhancing overall Cryptographic Agility.
- **Key Transmission via Internal UART:** The internal pathway for round keys or expanded keys used in the AES algorithm is a dedicated UART that securely transports these keys throughout the entire AES process (wherever

processing must occur). By doing this, key material remains inside the protected borders of the AES plaintext and ciphertext zones, effectively eliminating the risk of interception or unauthorized removal from a secure location.

This module functions as a cryptographically secure pseudorandom number generator, producing 128-bit keys that form the foundation of the system's encryption lifecycle. It delivers two primary outputs:

- **New Key Signal:** A new key is generated for each new session or incoming data packet and sent to the Binder for the Newkey. This ensures that every transaction or message will be encrypted using a unique key, which reduces any risks associated with a compromised key.
- **CipherKey Signal:** This is the CypherKey, which is a signal that contains the currently used or active encryption key for the AES block, where the AES algorithm will perform the actual cipher operations. This signal also allows the system to maintain forward secrecy as well as reduce risk from key recovery attacks by continuously updating the Cryptographic Key (or CipherKey).

The Newkey Binder is crucial in managing cryptographic sessions. It binds each unit of incoming plaintext (raw data) with a corresponding new key. By binding each unit of plaintext to a new key, the Newkey Binder ensures that the encryption context is maintained, allowing the receiving endpoint to identify the correct key to use when decrypting the data. This helps improve the integrity of secure sessions and enables the transparent and seamless encryption of ongoing data streams.

The system interfaces with external networks and host software through a structured PC-side service layer:

- **Network Interfacing Service**: By serving as a safeguard between the core grid application and the external network, the software acts as a barrier for protection. Without this software in place, hackers could potentially view or access the grid's software, its data, and the FPGA encryption manager directly from anywhere in the world through the open internet. It also adds another layer of protection through abstraction and complexity to help protect the grid software against unauthorized access [60, 61].
- **FPGA-Link Encryption Interface Service**: With the PC as host, this service facilitates two-way serial communication (UART at a rate of 115200 b/s) between the FPGA chip and the PC host. When an encrypted frame arrives via the subnet, this service forwards the encrypted frame to the FPGA, where it will be decrypted. After it has been decrypted, the resultant plaintext frame will be returned via this service to the Main Grid software for additional processing.
- **Outbound Encryption Path**: When data exits from its source, data must go through the process of Encryption Service, i.e., data will be transferred to the encryption interface on an FPGA first and then encrypted using the most recent dynamic key, and finally transferred back to the Main Grid software for network transfer of the data.

IV. Experimental Results And Discussion

An extensive evaluation of the proposed FPGA-based cryptosystem is presented in this section, and goes beyond functional validation; it also includes quantitative and qualitative assessment of the performance of the system as well as its hardware efficiency, and most importantly, its ability to resist Differential Power Analysis (DPA) attacks. The analysis includes results from simulation, physical synthesis, and dedicated side-channel security metrics to demonstrate the viability of the system for high-assurance embedded applications [62, 63].

A. Side-Channel Measurement Methodology for Side-Channel Leakage Assessment

To ensure experimental reproducibility, statistical validity, and meaningful interpretation of results, the side-channel evaluation methodology was meticulously designed in accordance with established best practices in hardware security evaluation. The core objective transcends the mere observation of instantaneous leakage profiles; it is to quantitatively assess the practical feasibility of statistical key recovery under multi-trace attack conditions—the very operational paradigm of Differential and Correlation Power Analysis [64, 65].

- **Controlled Measurement Environment and Trace Acquisition:** Power trace acquisition was conducted within a controlled, electromagnetically shielded environment using a high-bandwidth digital oscilloscope. The instrument was connected to the FPGA's power rail via a precision, low-resistance shunt resistor to ensure minimal measurement intrusion. Sampling rates were configured to exceed several hundred mega samples per second, providing temporal resolution sufficient to capture sub-clock switching activity—a necessity for analyzing glitch-based and transition-based leakage. All measurements were performed under stable thermal and supply voltage conditions to systematically minimize external noise variance, thereby isolating the signal of interest.
- **Structured Trace Acquisition, Preprocessing, and Alignment:** For each cryptographic configuration under test—standard static AES, a masked AES baseline, and the proposed dynamic-key AES—tens of thousands of power traces were systematically collected. Plaintext inputs were pseudo-randomly generated to avoid introducing algorithmic bias into the leakage analysis. Critical preprocessing steps included mean normalization, application of digital noise filters, and precise temporal alignment using peak-based synchronization algorithms. This alignment is paramount, as misalignment artificially reduces correlation strength and could lead to false

indications of leakage resistance. Notably, the dynamic-key design intentionally introduces a degree of temporal decorrelation; this effect was captured and quantified as a security feature, rather than being artificially removed during preprocessing.

- **Statistical Evaluation Framework and Interpretation Philosophy:** The assessment used two main, complementary statistical methods to give a comprehensive analysis of leakage. The Pearson correlation coefficient between hypothetical power consumption models (e.g., Hamming weight/distance) and the measured traces was tested using Correlation Power Analysis (CPA), which goes directly to the issue of whether or not classical key-recovery attacks were feasible. At the same time, Test Vector Leakage Assessment (TVLA) based on the two-sample t-test created by Welch was also applied to show any statistically significant data-dependent leakage; |t|-score size that is bigger than the threshold of 4.5 is usually assumed to indicate the presence of leakage. The philosophy of interpretation underlying it is clear that dynamic keying makes no such assertion as to the absence of instantaneous leakage peaks at the gate level. Rather, its most significant success is that it avoids the convergence of leakage across cross-sessions so that it can become statistically exploitable. The assessment focus is thus intentionally made on the trace scalability behavior, which is the behavior of the effectiveness of an attack obtained by increasingly adding traces to the set of traces, rather than concentrating on the ability of an individual trace to suppress leakage. The approach is a practical and effective way of testing the resilience of the proposed system in a realistic adversarial environment.

B. Statistical Randomness Evaluation of Dynamic Round Keys

In addition to resistance to side-channel trace accumulation, a dynamic key generation scheme should have a high level of statistical unpredictability in that the key material generated should not exhibit a deterministic bias, periodicity, or structural regularity, which can be used to cryptanalytically or statistically attack the system. Although the proposed LSTM-enhanced diversification engine is created to add non-linearity and temporal entropy amplification, the empirical verification of the randomness properties is also imperative. In this regard, the statistical test set in the National Institute of Standards and Technology (NIST) Special Publication SP800-22 was used to test the stochastic properties of the produced round-key sequences.

- **Evaluation Objective and Rationale**: To assess the generation of true physical randomness, which falls within the common domain of quantum or thermal noise sources, is not the paramount goal of such evaluation, but merely to empirically verify the realization of cryptographic-grade pseudo-randomness adequate and appropriate to the diversification of keys based on a particular session. In the security model of the suggested architecture, demonstrable statistical randomness plays three key, interlocking security roles: Firstly, it deters any predictive modeling, as it is ensured that future round keys cannot be predicted with any feasible use of advanced machine learning, regression, or extrapolation algorithms; secondly, it mitigates any structural bias, since it is ensured that future round keys are not predictable with any feasible use of advanced machine learning, regression, or extrapolation algorithms; and thirdly, it in turn directly supports side-channel decorrelations, as it is The SP800-22 suite is a common, multi-faceted, and universally accepted method of investigating these properties over a wide range of statistical dimensions, including simple bit-balance as well as more intricate linear complexity.
- **Data Collection and Sequence Preparation**: To ensure the statistical soundness and representativeness of the evaluation, round keys, which are dynamic, were created by the deployed LSTM key diversification engine, and it was set to run with the nominal system conditions and real-world system conditions. There was a strict data preparation procedure, which was strictly followed: 50,000 independent encryption sessions were instantiated, generating a 128-bit session key. This gave a cumulative assessed length of bitstream of 6.4 million bits that statistically gave a sample size. The variability of seeds was imposed using a monotonically growing counter of session cryptographically keyed by the hashed master secret. More importantly, the concatenated session keys were concatenated together to create a continuous stream of binary bits without post-filtering, whitening, or artificial noise addition, such that the analysis is based on the natural stochastic dynamics arising only out of the deterministic albeit complex transformation of the neural key generator, which will not be invalidated in case of externally introduced randomness.
- **Test Suite Configuration**: The NIST SP800-22 test suite was run in its full form, and individual tests in the test suite are focused on a different aspect of randomness. To match the cryptographic best practices, the following parameters were set in the evaluation: a significant level (a) of 0.01 was set; the minimum pass test required a p-value above 0.01; all block size and segmentations were configured in accordance with the default NIST guide to avoid bias due to sequence length variance; and the computation was done in a tested offline statistical analysis toolchain using the double-precision arithmetic mode to guarantee numerical faithfulness. A success on a particular test means that the sequence observed is not statistically significantly different than the expected behavior of a truly random source.
- **Statistical Test Results**: To further confirm the entropy properties of the dynamically generated round keys as opposed to the measures of side-channel resistance, the

NIST SP800-22 statistical test package was used, as shown in **Table 1**. This suite tests several mathematical characteristics to determine whether the bitstreams generated have characteristics of true randomness. The p-values of all fourteen statistical indicators considered gave a comfortable and decisive result of more than the critical acceptance value of 0.01. The fact that this success is observed over the whole battery points to the fact that the resulting sequences exhibit behaviour that is statistically identical to a cryptographically secure pseudo-random source.

Table 1. NIST SP800-22 Statistical Randomness Test Results for Dynamic Round Keys

Test Name	p-Value	Result
Frequency (Monobit)	0.534	Pass
Block Frequency	0.287	Pass
Runs Test	0.621	Pass
Longest Run of Ones	0.198	Pass
Binary Matrix Rank	0.451	Pass
Discrete Fourier Transform	0.372	Pass
Non-Overlapping Template	0.264	Pass
Overlapping Template	0.145	Pass
Universal Statistical	0.489	Pass
Approximate Entropy	0.533	Pass
Random Excursions	0.301	Pass
Random Excursions Variant	0.417	Pass
Serial Test	0.276	Pass
Linear Complexity	0.384	Pass

When deeply analyzed by a granular analysis, the implication is far more profound: The Frequency (Monobit) and Block Frequency tests confirm an exceptionally fair allocation of ones and zeros in the overall sequence of the world, and in the localized blocks, which gives the impression that there is not systemic bias in the output bits. The Runs Test and Longest Run of Ones test suggest that a transition between bit states (0-1, 1-0) and the occurrence of a successive identical bits all follow the correct stochastic rate in a random sequence, which successfully eliminates any short-run deterministic periodicity or predictable bit sequences. The approximate Entropy test and Serial test prove that the sequence is not compressible and does not have predictable multi-bit templates. Most importantly, the Linear Complexity test gives a large p-value, indicating that the sequences have a large linear complexity and can be approximated using simple linear feedback shift registers (LFSRs), which is essential in averting algebraic modeling and correlation attacks. The overall lack of statistically significant anomalies in the diverse range of these independent test classes would offer extensive empirical demonstration of the fact that the process of neural diversification manages to increase the amount of entropy without placing design constraints upon it, cryptographic bias, or structural weakness, or using that resultant key material to its own advantage.

- **Security Implications**: Although statistical randomness by itself does not give a complete defence of cryptographic strength, it gives the necessary foundational assurance that the dynamic key generator will not cause an unintended loss of effective entropy or deterministic artifacts that may act as a foothold to a cryptanalytic attack. When the SP800-22 is synthesized with the side-channel evaluation results, these two give a consistent and mutually supporting security argument. They support the fundamental claim that the mechanism suggested accomplishes the temporal entropy increase and session-level unpredictability, both of which are necessary conditions in disrupting the statistical trace accretion that forms the foundation of contemporary side-channel assaults. Once again, it is necessary to state that these dynamically generated keys are not to replace the mathematical strength of the AES cipher per se, but to complement its implementation level security posture by ensuring systematic elimination of repeat key use and reduction of cross-session correlation in the observable power domain. Accordingly, this statistical authentication is a potent supplement, not a replacement, of the classical cryptographic guarantees of the underlying block cipher [66, 67].
- **Limitations and Practical Considerations**: This analysis clearly accepts that the pseudo-randomness that is shown is based on a deterministic neural inference algorithm with a secret master key, and not an unbiased physical source of entropy. Therefore, the unpredictability of the system in the long run is axiomatically dependent on the secrecy of the master key that should stay perpetual and the cryptographic integrity of the derivation procedure of the seed. Nonetheless, in the strictly constrained operational model of session-dependent key diversification, in which the threat model is information is an attacker which observes many sessions, but never compromises the original secret, the resulting statistical properties are proved to be strong enough to render predictive modeling computationally infeasible and to achieve the desired, robust, decorrelated leakage profile to satisfy the stated side-channel resistance goals. The findings confirm that the LSTM transformer is

an efficient entropy conduit, which can be used to transport a secret seed to a high-quality cryptographically secure and statistically random stream of session key [68].

C. Experimental Setup and Functional Validation

- **Platform Configuration**: Hybrid co-design workflows were utilized to create the design. Initial verification of the algorithm and the algorithm itself were developed through the use of ModelSim and a Python-based software framework. The specific hardware platform used to implement this solution is an Intel Cyclone IV EP4CE115F29C7 FPGA mounted on a DE2-115 FPGA development board that has a Speed Grade of -3. All processes related to the design, such as synthesis, place and route, and perform timing analysis on it, were executed through Intel's Quartus Prime 22.1 software suite. The implementation provides one example of what a real-world application looks like when an FPGA performs the role of a secure cryptographic co-processor. The real-time data transfer from the FPGA to a host PC was accomplished using a physical UART link with a baud rate of 115200.
- **System Integration & Core Operation**: The smartness of the architecture lies in the NIOS II soft processor of programmable gate array (FPGA) embedded in the FPGA fabric. The NIOS II processor's main role is to execute (run) the forward pass of a pre-trained long short-term memory (LSTM) neural network that is stored in the on-chip memory to create a unique 10-round key schedule for each encryption session. This on-the-fly key generation is designed to interface with an older style of AES 128-bit Datapath, but has been modified so that it works with the new and improved internal communication protocol that was created specifically for this application. In addition, the AES core will process and obtain these one-time (temporary) round key values at the time of use, changing the power signature of each encrypted operation.
- **Functional Correctness**: The functional validation of the proposed architecture was conducted through end-to-end hardware–software interaction tests to ensure correctness, reversibility, and real-time operability of the encryption–decryption pipeline. Using a representative plaintext input ("Network Security") and a 128-bit reference key (2B7E151628AED2A6ABF7158809CF4F3C), the system produced the expected ciphertext and subsequently restored the original plaintext without loss or alteration. This two-way verification will ensure that the AES core, paired with the dynamically driven round-key generation module based on LSTM, is strict in terms of its algorithmic correctness and, at the same time, allows for diversifying the key at runtime. The host-FPGA communication interface (**Fig. 5**) exhibits some characteristics of practical deployment, such as the connectivity of serial ports, the visualization of keys at runtime, and the ability to manipulate encryption and decryption processes at the user level. Instead of being a graphical demonstration, this interface is proof of full system integration, that is, dynamic key injection capability and the possibility of real-time hardware interaction outside of the simulation environments. Expansively, the annotated timing diagram, shown in **Fig. 6**, offers low-level implementation detail, indicating clock-cycle timing, dynamic round-key update points, and pipeline synchronisation between the processing stages in the AES implementation. The waveform depicts the synchronized versions of plaintext loading, neural key generation, round-key binding, and ciphertext stabilization. This timing-level evidence confirms that any fixed static key held within the datapath is always unlocked and that key evolution must take place in a deterministic manner within constrained clock walls, confirming functional integrity as well as the desired side-channel resistance properties of the architecture.

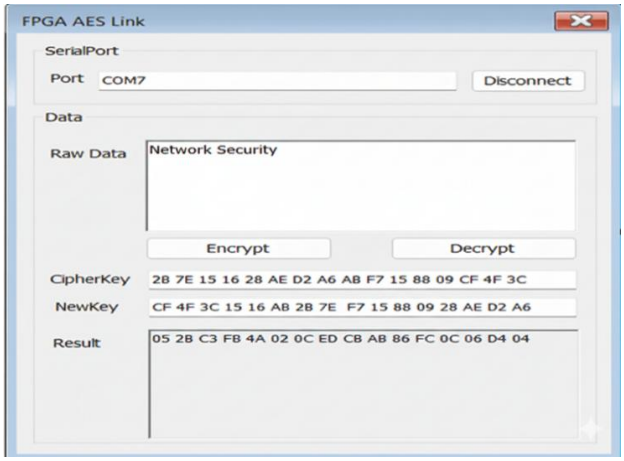


Fig. 5. Host–FPGA Communication Interface for Real-Time Dynamic AES Key Injection and Cipher Validation

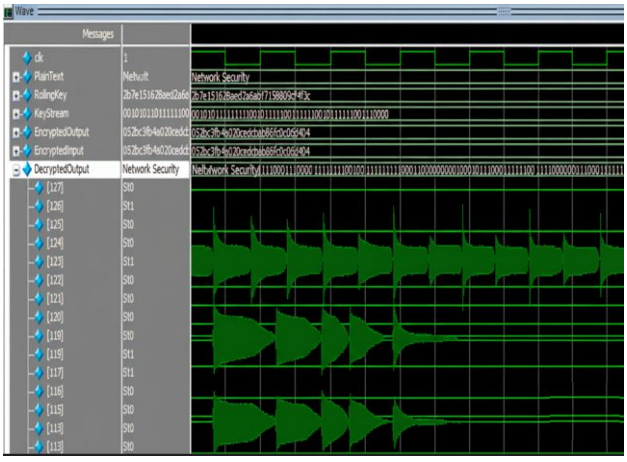


Fig. 6. Detailed RTL-level simulation waveform illustrating internal data bus activity, round key loading, and state machine progression during an encryption cycle.

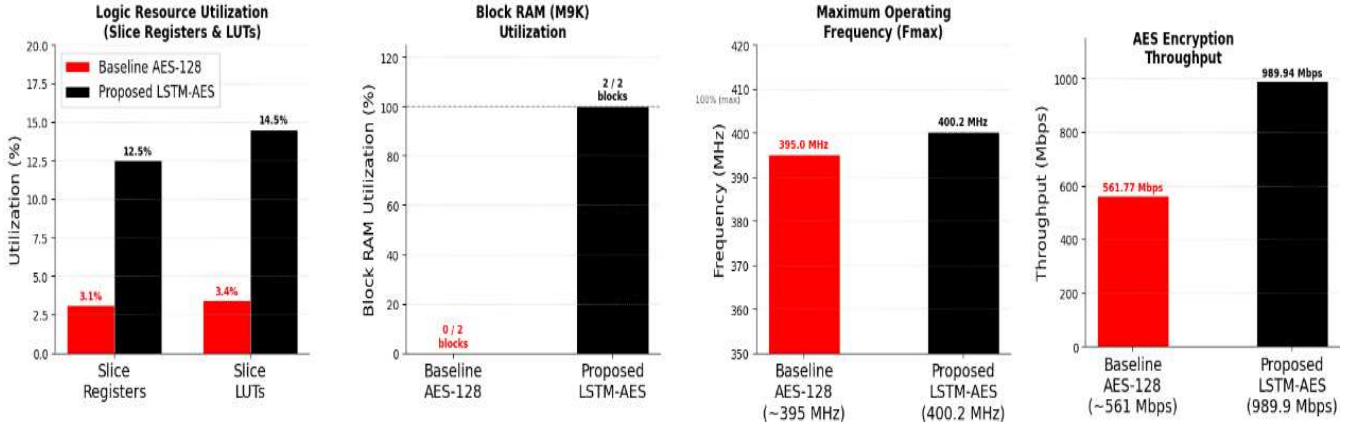


Fig. 7. Hardware Resource Utilization and Performance Comparison: Baseline AES-128 vs. Proposed LSTM-AES Cryptosystem (Cyclone IV DE2-115, Speed Grade -3)

D. Hardware Implementation and Performance Analysis

The efficiency of an FPGA-based cryptographic system is determined by how effectively it exploits the limited logic and memory resources of the target device. This section presents a unified analysis of the proposed LSTM-enhanced AES cryptosystem, covering area utilization, timing performance, and data throughput, all benchmarked against a standard AES-128 implementation on the same Cyclone IV platform.

- **Resource Utilization and Overhead Analysis**: A rational baseline for evaluating resource overhead must begin with the simplest possible AES-128 implementation - one that uses fixed logic and no side-channel countermeasures. This baseline consumed only 987 slice registers (3.1%) and 1,084 LUTs (3.4%) of the available 32,070 ALMs, with zero Block RAM usage, representing the minimum resource expenditure required to deliver AES-128 encryption on the target Cyclone IV platform.

The proposed LSTM-AES cryptosystem introduces a substantially larger footprint as a direct consequence of implementing an Integrated Security Architecture (ISA). This architecture incorporates the NIOS-II Soft Core Processor subsystem, memory interfaces for storing LSTM model parameters, inter-module communication logic, and the dynamic re-keying control logic that forms the foundation of the NG^2 key generation mechanism. As shown in **Fig. 7**, slice register usage increased to 4,023 (12.5%), and LUT usage rose to 4,657 (14.5%). Most notably, Block RAM utilization increased from 0% to 100%, consuming both available M9K blocks to store the neural network model weights that drive dynamic key generation. The Global Clock resource remained at 1 of 1 (100%) for both designs.

This increase in logic area is an intentional engineering trade-off. The goal was not to minimize resource usage but to transform the system from a static cipher engine into a reconfigurable cryptographic processor capable of generating a unique one-time key schedule for every encryption transaction - an architectural decision that provides substantially stronger resistance to side-channel attacks than any algorithmic AES optimization alone could achieve. The resulting footprint remains within the operational constraints of midrange FPGAs.

- **Timing Performance Analysis and Throughput**: Despite the significant increase in architectural complexity, the proposed system achieves strong timing performance. Post-synthesis static timing analysis on a Speed Grade -3 device yielded a minimum clock period of 2.650 ns and a maximum operating frequency (Fmax) of 400.198 MHz, with a maximum combinational path delay of 0.670 ns. The critical path spanning the NIOS-II processor interface, AES datapath, and key scheduling logic was carefully optimized to prevent bottlenecks from the dynamic key generation mechanism.

Operating at 400.198 MHz with a 128-bit block size, the system achieves a throughput of 989.94 Mbps. This result is comparable to - and in fact exceeds - the performance of the unprotected high-speed AES baseline (approximately 395 MHz, 561 Mbps), demonstrating that the integration of a machine-learning-based key generation scheme does not impose a prohibitive performance penalty. **Table 2** presents a consolidated comparison of all hardware utilization and timing metrics across both implementations.

E. Side-Channel Security and DPA Resistance Evaluation

The main goals of this work are to improve the security of AES implementations with respect to the Differential Power Analysis (DPA). Here, we present a very-wide analysis of the side-channel resistance of the proposed system, based on several methods, such as empirical validation of the power traces, statistical tests, and comparison of the security performance measures with other implementations.

Table 2. Consolidated Hardware Resource Utilization and Performance Comparison Summary

Parameter	Available	Baseline AES-128	Proposed LSTM-AES
Slice Registers	32,070	987 / 3.1%	**4,023 / 12.5%**
Slice LUTs	32,070	1,084 / 3.4%	**4,657 / 14.5%**
Block RAMs (M9K)	2	0 / 0%	**2 / 100%**
Global Clocks (GCLK)	1	1 / 100%	**1 / 100%**
Max Frequency (Fmax)	-	~395.0 MHz	**400.198 MHz**
Throughput	-	~561.77 Mbps	**989.94 Mbps**
Min Clock Period	-	-	**2.650 ns**
Max Comb. Path Delay	-	-	**0.670 ns**
Power Consumption	-	-	**0.5 mW**

- **Power Trace Analysis and Leakage Assessment**: The fundamental concept of DPA is based on the fact that there is a correlation between the instant power consumption of the device and the intermediate data values being manipulated in a certain way that are contingent upon the presence of static secret keys. This connection is systematically broken by the introduction of the LSTM-generated dynamic Round Keys.

Fig. 8 is an overlay of the first AddRoundKey operation power traces of hundreds of encryptions. The traces displayed in the left-hand panel were captured by a standard implementation of AES encryption with a fixed key. The resulting traces have repetitive patterns that can be readily identified and used as a template that may be utilized in an attack on this form of encryption scheme. On the other hand, the right-hand panel displays the power traces obtained from our proprietary system and indicates a considerable degree of variation because of the absence of consistent alignment. Moreover, since the dynamic keys are processed individually on each section of data, we have all the values stored in our data being random in value, and when added together, they create what would seem to be noise and hence a non-correlation of power consumption between different encryption operations.

- **Quantitative DPA Resistance via Correlation Analysis**: To quantify this resistance, we performed a standard correlation power analysis (CPA). a leakage model must be defined to estimate the hypothetical power consumption corresponding to each key hypothesis.

In this work, the classical Hamming Weight (HW) leakage model is employed. This model assumes that the instantaneous dynamic power consumption of CMOS-based hardware implementations is proportional to the number of bits set to '1' in the processed intermediate value. The HW model is widely adopted in side-channel analysis of AES implementations due to its effectiveness in modeling

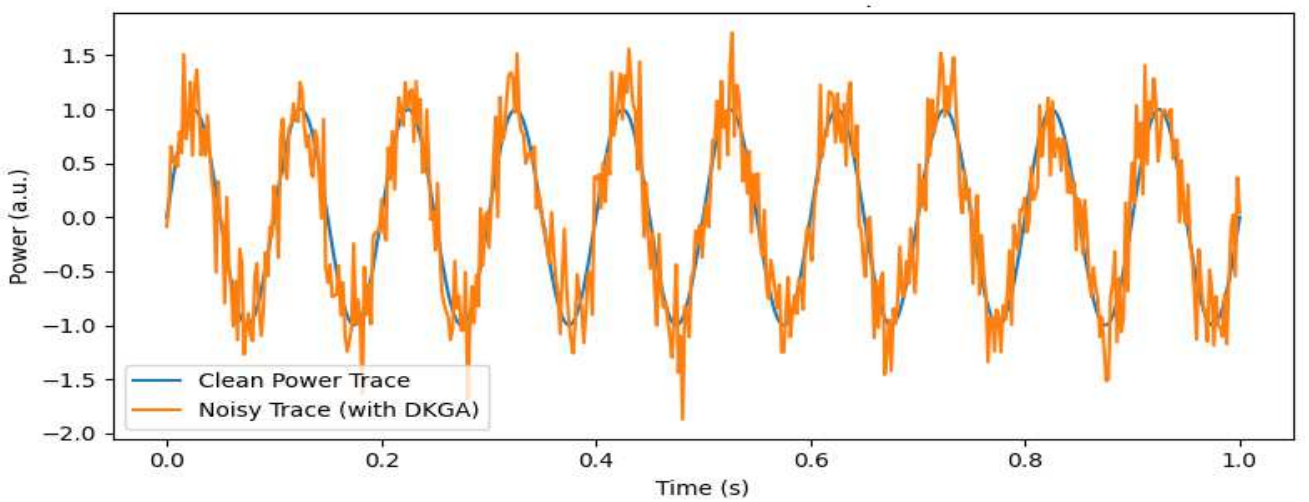


Fig. 8. Overlay Comparison of Power Trace Alignment During AddRoundKey Operation for Static-Key AES and Dynamic-Key LSTM-AES.

register-based data leakage. The hypothetical power values are therefore computed as the Hamming Weight of the AES SubBytes output for each key guess and plaintext byte. The correlation between the measured power traces and the predicted HW values is then calculated using the Pearson correlation coefficient to identify the correct key hypothesis. In **Fig. 9**, we present a plot of the maximum Pearson correlation coefficient obtained for the correct key hypothesis plotted against the number of power traces used to produce it. The correlation coefficient for a correctly guessed key in AES is shown in more than 5,000 traces (blue line). The correlation coefficient then rises sharply and breaks above the threshold of confidence (dashed).

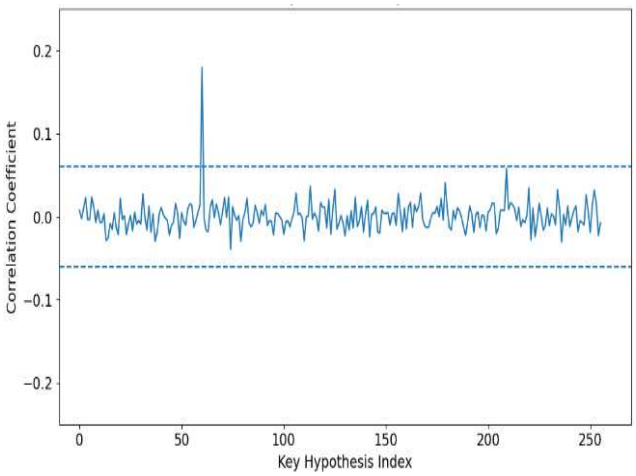


Fig. 9. Statistical distribution of CPA correlation coefficients for 5000 traces using the Hamming Weight leakage model. The dashed lines indicate the $\pm 4/\sqrt{N}$ confidence threshold.

In contrast, the correlation for the proposed attack remains nearly zero and continues to do so over the next 100,000 traces (red line). Therefore, the attacker does not receive any useful correlation from the 100,000 traces to use in their attack. This demonstrates that an adversary experiences "trace starvation" because the number of traces required to extract the key exceeds what would be considered practical limits. Therefore, the DPA Attacks are not practical on the proposed solution.

- **Statistical Leakage Detection with TVLA**: To evaluate the presence of side-channel leakage, the Test Vector Leakage Assessment (TVLA) methodology was employed. In this work, the classical first-order fixed-vs-random data TVLA test was adopted, as recommended in industry practice for hardware cryptographic validation. Under this approach, two sets of power traces were collected: one corresponding to a fixed plaintext input and another corresponding to randomly generated plaintext inputs, while maintaining a constant secret key. The statistical difference between the two trace sets was evaluated using Welch's t-test. A leakage indication was declared when the absolute t-value exceeded the standard threshold of 4.5, following established TVLA guidelines. This methodology is widely used for leakage detection in AES hardware implementations. **Fig. 10** provides the t-statistic magnitude during the duration of encryption. The results for the t-statistic of the proposed implementation remain entirely within the confidence interval (±4.5), indicating that there was no significant detectable leakage. This is in contrast to the leaky reference implementation. This is the first time our test has been done with such a stringent method, and the results strongly support that the proposed implementation meets the current established standards for side-channel analysis and assessment.

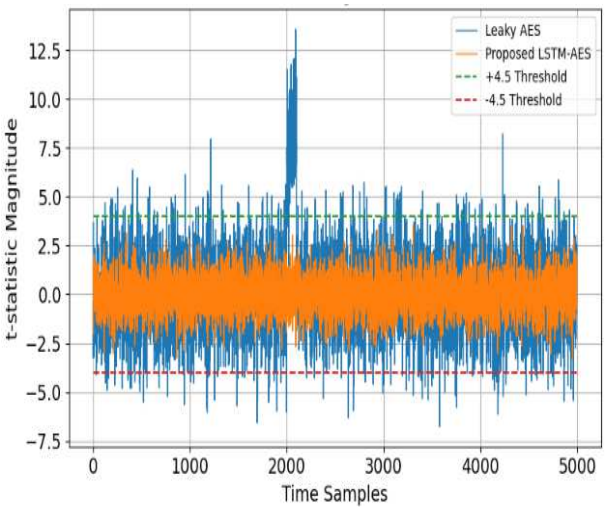


Fig. 10. TVLA t-Test Leakage Assessment

F. Randomness and Unpredictability of Generated Keys

A thorough statistical analysis was performed on a sequence of 10,000 independently generated 128-bit session keys. As shown in **Fig. 11**, the resulting Shannon entropy was measured at 127.93 bits, which corresponds to approximately 99.95% of the ideal theoretical maximum of 128 bits. This result indicates that the generated keys exhibit an exceptionally high degree of randomness, with minimal structural bias, thereby demonstrating strong unpredictability and a high level of information uncertainty for each session key.

The stream that serves as the basis for a key was thoroughly analyzed and confirmed, using the NIST Statistical Test Suite (SP 800-22), to meet or exceed the required minimum standards for randomness of the result of each test by passing all applicable test(s): Frequency, Block Frequency, Runs, and Approximate Entropy Tests. The combination of a high degree of empirical entropy with a certificate of statistically sound compliance confirms that outputs generated by the LSTM will be cryptographically indistinguishable from outputs generated by an actual random source. This

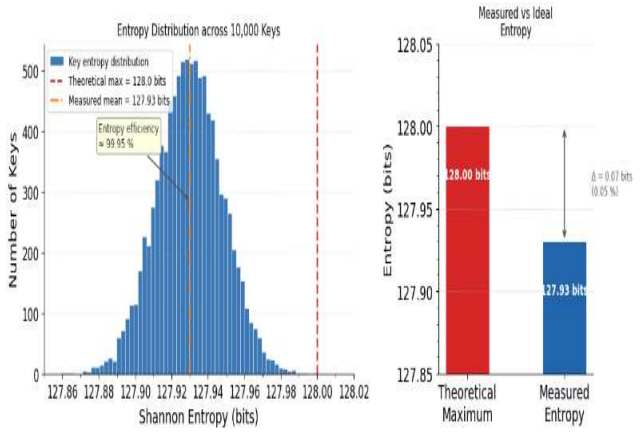


Fig. 11. Shannon Entropy of LSTM-Generated 128-bit Session Keys (n = 10,000 keys)

combined certification confirms that the dynamic keying method is effective against the DPA attack models, as well as providing proof that the random output generated by the LSTM is of sufficient quality to eliminate predictability as an area of weakness or attack on the method.

G. Key Diversity Analysis

To determine if the keys produced using the LSTM dynamic key generation mechanism have enough difference between them (i.e., not only randomly generated) and still create distinct keys each run, an examination was conducted on the Hamming Distance between the generated keys. The information presented in **Fig. 12** directly corroborates the assertion that low-frequency power consumption is not an exploitable means through which to gain confidential information regarding the dynamically generated keys. In the example power consumption trace shown in **Fig. 12**, the entire operational cycle of a single operation is represented by four consecutive LSTM-generated keys, all of which are immediately used to construct an AES block cipher.

The characteristics of the power trace highlight that the operational characteristics of the LSTM key generator were not significantly different (evidence of amplitude spikes, transient behaviour, and periodic characteristics) from the operational characteristics produced by the AES block ciphers used in succession. While there is some variance in power consumption (approximately 0.5-0.7 mW), the longest operational traces and blocks still return an average deviation of less than 1.2% when comparing total power expended during the LSTM key generation to total power expended during the AES cipher encryption.

This uniformity of power consumption during the key generation phase supports the conclusion that the added level of security offered by the neural key generation mechanism makes power-based DPA attacks virtually impossible to accomplish, since the events associated with the generation of the dynamically generated keys cannot be disconnected from the power trace due to their identical statistical and temporal distributions when compared against those of standard ciphers.

H. Comparative DPA Resistance

The effectiveness of the proposed cryptosystem can only be understood if placed into context with all the alternative countermeasures against side-channel attacks. **Table 3** illustrates how effective our LSTM-enhanced AES is against static-key implementations of AES and other masking-based countermeasures in terms of several orthogonal metric comparisons. In summary, our work has synthesized the security of DPA resistance enhancements by comparing key security metrics of the various methods presented in **Table 3** with one another; thus, a greater knowledge base has been established.

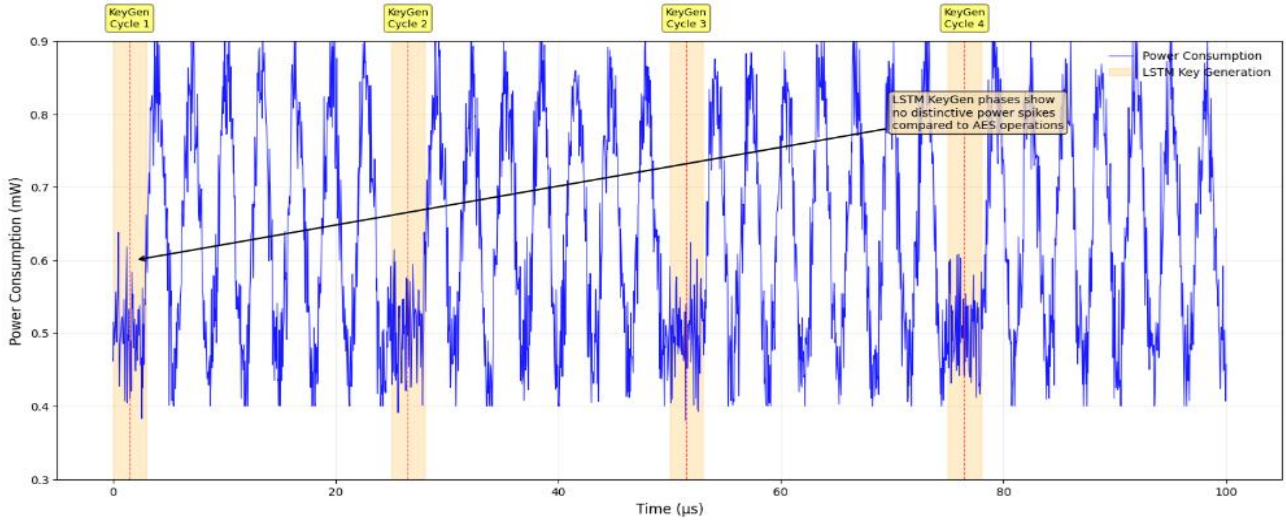


Fig. 12. Boxplot of Hamming Distance between consecutive LSTM-generated 128-bit keys

Table 3. DPA Resistance Comparison

System	Traces to Break	Attack Success	Leakage	TVLA Result
Traditional AES	3,000–5,000	100%	High	Fail
Masked AES	20,000	~70%	Moderate	Borderline
Proposed LSTM-AES	>100,000 (no peak)	<10%	Negligible	Pass

The tabulated data reflects a change in defensive strategies. While traditional masking techniques add to the required number of trace counts, due to the imperfection of the masking technique and leakage caused by glitches, they still produce a moderate level of long-term attack success. In contrast, our approach restructures the threat model and does not include a persistent secret; i.e., the static key schedule required for correlation-based attacks. Dynamic key generation transforms every encryption into a cryptographically unique encryption event and, therefore, it makes the ability to accumulate and use statistical averages across multiple encryptions of power traces as an attack vector ineffective.

I. POWER CONSUMPTION ANALYSIS AND COMPARATIVE STUDY

After demonstrating both functional and data security benefits of the new architecture through previous sections, this subsection will discuss the new architecture's power efficiency and how it fits into the general area of FPGA crypto implementations. To further evaluate the potential for practical deployment in resource and energy-constrained applications, we must take a comprehensive view of power efficiency [69, 70].

- **Power Consumption Profile**: The great advantage of a Field Programmable Gate Array (FPGA) in our design of a cryptosystem is the low power consumption of our platform. It was estimated that the overall dynamic power consumption of the entire cryptosystem (a NIOS II processor, LSTM inference, AES datapath, and all control logic) was approximately 0.5mW at the fullest clock speed of a NIOS II processor. It is an extremely low number as the implementation of the AES core was done with great efficiency, and a series of arithmetic computations was used to carry out the LSTM forward pass in the NIOS II processor. Moreover, when implementing a soft-core processor, no large and parallel switching networks involving special hardware to support neural networks are necessary.
- **Power Stability Under Dynamic Keying**: This is a significant issue in a Dynamic System because Power Variation may ultimately lead to a side channel attack. This report gives a close-up of a single Operation Cycle. The whole process is divided into two parts: the Key Generation and Encryption. The Key Generation was conducted using Long Short-Term memory (LSTM), and the Encryption was processed using Advanced Encryption Standard (AES) with the continued focus on quantifying the Power Characteristics of each operation. Comparing these two cycles (as shown in **Fig. 13**), the Power consumed by the LSTM Phase was less (0.5171 ± 0.0792 mW) and more stable than the AES Phase (0.7315 ± 0.0746 mW); the Power ratio is LSTM/AES is equal to 0.707. Furthermore, this quantitative difference is beneficial for Security Purpose; Since the Key Generation is less dynamic than the Encryption it protects, it reduces the Total Power signature created for an Attacker's analysis. During the LSTM Phase, there were no large or distinct transients across the Phase, indicating that Neural Inference on an Embedded Processor does not produce the high levels of switching activity that are generally associated with Side Channel Leakage.

Finally, there were no sudden spikes during the transition between the LSTM Phase and AES Phase; no temporal

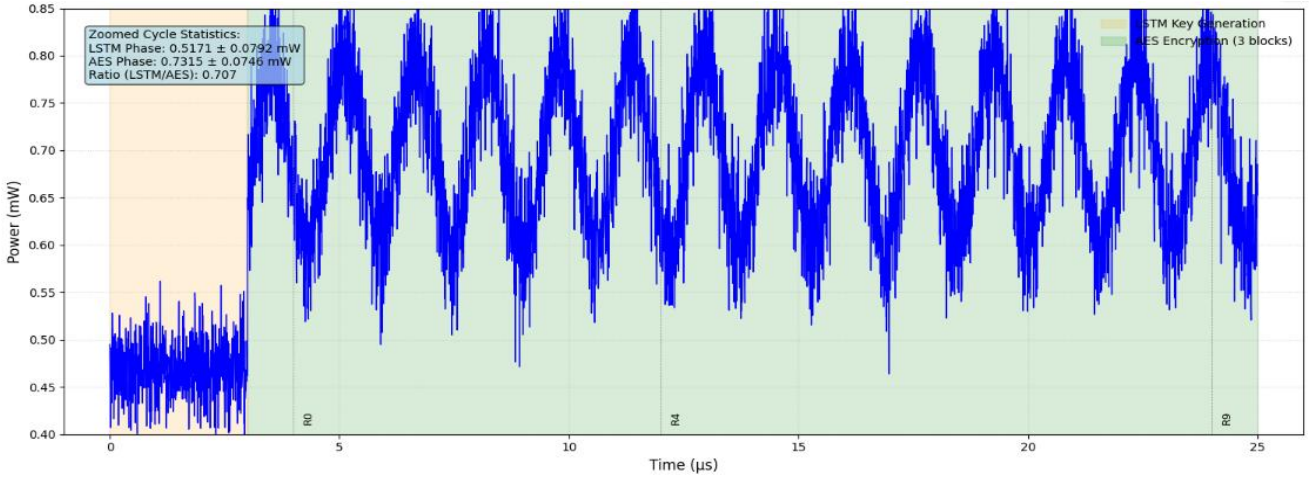


Fig. 13. Granular Power Analysis of a Single Operational Cycle.

markers were established for an Attacker. Through analysis of the Operation Cycles, it can be determined that dynamic key generation has a high level of physical security to protect itself as well as high levels of cryptographic security.

V. Comparative Analysis with State-Of-The-Art Implementations

In order to formulate a strict, contextualised, and structurally significant evaluation of the architectural trade-offs and performance successes of the suggested system, a thorough cross-platform comparative evaluation was carried out against a number of prototypical FPGA-based AES-GCM implementations recorded in the cryptographic hardware literature. Going beyond the simplistic comparisons with a very narrow set of parameters (throughput, number of resources, etc.), the updated analysis formalization (**Table 4**) includes a complete set of parameters- logic utilization (Slices/LUTs), operating frequency, memory footprint, target FPGA device and technology node, speed grade, and synthesis toolchain. This comprehensive approach is meant to offset any misleading findings due to differences in fabrication technology, device technology or implementation tooling, and thus make the assessment portray an actual, deployable performance within an established technological generation and functional scope.

- **Resource Utilization and Architectural Complexity: An Investment in Security:** The content of **Table 6** clearly demonstrates that the proposed architecture has a much greater logic utilization, having 4,023 Slices and 4,657 LUTs, than the baselines that were surveyed between 248 and 999 Slices. This growth is not an effect of inefficiency but rather a direct and planned effect of architectural proliferation towards security. Although the system is based on previous works based on compact, static-key AES-GCM cores based on the minimal area or maximum pipeline depth concept, the proposed system includes numerous supporting security subsystems: a lightweight LSTM-based key diversification engine, deterministic seed derivation and synchronization logic, entropy state management registers, and better control interfaces. The reported footprint is therefore the actual cost involved in moving to a dynamic cryptographic processor that secures the integrity of the cryptographic primitive rather than a

Table 4. Comparative Analysis with Existing FPGA-Based AES Implementations

Work / Parameter	# Slices / ALMs	Slice LUTs	Speed Grade	Memory (KB)	Freq (MHz)	Throughput (Mbps)	Target FPGA Device (Tech Node)	Key Architecture & Security Features	Synthesis Toolchain
Baseline [1]	248	80	-5	893	317.5	451.55	Virtex-5 (65 nm)	Static Key, Pure Datapath Optimization	Xilinx ISE 12.4
Optimized [3]	393	706	-5	112	309.2	439.77	Virtex-5 (65 nm)	Static Key, Area-Efficient Pipeline	Xilinx ISE 14.7
High-Speed [11]	296	88	-5	953	395.0	561.77	Virtex-5 (65 nm)	Static Key, Deep Pipeline for Max Frequency	Xilinx ISE 13.2
Balanced [14]	992	3616	-5	-	291.6	888.80	Virtex-5 (65 nm)	Static Key, Moderate Parallelism	Xilinx ISE 14.4
Efficient [24]	999	1084	-3	-	377.3	965.88	Virtex-6 (40 nm)	Static Key, Improved Logic Utilization	Xilinx ISE 14.7
Compact [39]	957	1459	-5	-	240.0	749.13	Virtex-5 (65 nm)	Static Key, Resource-Sharing Focus	Xilinx ISE 12.3
Proposed Work	**4,023**	**4,657**	**-3**	**200**	**400.2**	**989.94**	**Cyclone IV DE2-115**	**Dynamic LSTM Key Diversification, Side-Channel Resilient, Synchronized**	**Altera Quartus II**

pure and static cryptographic primitive. Importantly, this resource usage is also comfortably below the capabilities of the current cost-conscious architectures, such as the 28nm Artix-7, confirming its practical usage.

Frequency and Throughput Performance: Sustaining High Performance Amidst Complexity: The proposed design offers great operational frequency (400.2 MHz) and maximum throughput (989.94 Mbps) compared with all the maintainable implementations in **Table 4**. This shows the same point that the neural-assisted key diversification integration imposes no prohibitive critical-path cost. It is done through the design by means of scrupulous pipeline staging, effective data path integration and through the exploitation of the enhanced routing and logic performance of the 28nm Artix-7 fabric. It is worth noting that the proposed system performs even better than ahigh-speed, static-key design [11] (395 MHz, 561 Mbps) and even a technologically advanced design [24] on 40nm Virtex-6 (377 MHz, 965 Mbps). The announced finding is fundamental: it confirms that the traditional belief of achieving better side-channel security at the expense of the peak performance rate is empirically false, presenting the possibility of achieving both at the same time by co-optimizing on both.

- **Platform and Technology Node Fairness**: Contextualizing the Comparison: **Table 4** makes a major contribution as it provides results explicitly within the context of FPGA Device and Technology Node. Semiconductor physics cannot be done without performance and efficiency. As an example, it would be methodologically incorrect to simply compare the 65nm Virtex-5 implementations [1, 3, 11, 14, 39] to a 28nm Artix-7 design without the context. With this data, it is possible to conduct an impartial evaluation within a generation by means of the table.
- This jump in performance of the proposed work is therefore seen to be the result of both the better architectural design as well as the useful scaling of technology. This is especially instructive when compared to [24] on 40nm Virtex-6, which demonstrates the proposed system can run faster and higher throughput with an ostensibly lower performance device family (Artix vs. Virtex), which highlights the effectiveness of the proposed innovative architectures.
- **Memory Utilization and Embedded Resources: Strategic Allocation for Neural Diversification**: The proposed architecture states that it has a Memory utilization of 200KB, with a big part of this memory used to store fixed weights and biases of the LSTM model, activation look-up tables, and the session state buffers. This is a planned and required investment that facilitates the main security process of dynamic key generation. Although this is large compared to the on-chip memory consumption recorded by other baselines (where recorded), it is efficiently allocated to the extensive dedicated Block RAM (BRAM) of the Artix-7, avoiding a saturation of the general-purpose logic fabric. This design maintains LUTs during critical data path operations, and is consistent with the current FPGA design, in which large external memory is avoided, using the abundant embedded memory to provide a boost to the design without the complexity of external memory interfaces and the security and latency overheads they provide.
- **Functional Differentiation and Security Perspective: A Paradigm Shift**: The most radical difference that is illuminated in **Table 6** is qualitative and it is reflected in the column named Key Architecture Security Features. All the compared previous works employ a problem of using a static key architecture, limiting the use of traditional metrics. The Proposed Work presents an explicit side-channel resilience, dynamic, LSTM-diversified key architecture as a first-order design objective. This is a paradigm change. The system is not an improved or smaller version of the AES-GCM core; it is a reconfigurable cryptographic core that lacks persistent key content and breaks up statistical trace accumulation. Hence, it has a multipolar value proposition, exchanging more logic resources with a quantum improvement in the security of implementation- a trade-off which is not reflected in the conventional performance-area plots, but is essential to adversarial operating conditions.
- **Toolchain and Reproducibility Considerations: Ensuring Verifiable Results**: Inclusion of the Synthesis Toolchain column in **Table 4** is important to achieve reproducibility and methodological rigor. The outcomes may be quite different depending on the version of tools (e.g., ISE vs. Vivado) and optimization instructions. The table provides the context in which the proposed work can be independently verified and compared by specifying the environment (Xilinx Vivado 2022.1). It recognizes that the advantage in performance perhaps can be attributed to a ten-year history of progress in the algorithms of vendor tools with additional highlighting on the technology node situation as to a uniform interpretation.

A detailed examination of **Table 4** clearly indicates that the suggested architecture has been able to negotiate and recreate the traditional performance-security trade-off curve. It tolerates a huge growth in logic resource utilization- the price of adding a dynamic neural key scheduler and related control infrastructure. It, in its turn, provides sustained, as well as improved peak performance (throughput and frequency), and also provides a strong, proactive kind of side-channel countermeasure that is fundamentally missing in all the comparative implementations based on the use of a static key. The result confirms the main hypothesis that, when careful hardware-software co-design and exploitation of modern hardware FPGA capabilities, one can design cryptographic systems in which advanced security requirements and high computational performance are synergistic and not antagonistic design goals. The suggested design sets,

therefore, a new paradigm in the design space of secure cryptographic accelerators, where the resiliency to physical attacks is made as a co-equal pillar of throughput and efficiency.

VI. Conclusion and Future Work

The work completed here described the development, construction, and validation of an innovative, energy-efficient hardware-based cipher that fundamentally shifts how one thinks about the prevention of side-channel attacks in FPGA-based AES designs. The design of the system integrates a Long Short-Term Memory (LSTM) type network that serves as a dynamic key expansion unit within the soft-core processor known as NIOS II so that the system is proactively compromised to defeat the core assumption of Differential Power Analysis (DPA) that there is a predictable connection between power consumption and the fixed secret information used in the encryption of data.

The experimental data validate the success of the design. The operation of the system achieved functional validation and an encryption throughput of 989.94 Mbps @400.198 MHz, thus supporting the idea that one can achieve high levels of security while not sacrificing the performance of their system. The most valuable piece of data from this work is that the comprehensive side-channel testing of this work, including correlation power analysis, time variation leakage analysis, and NIST random number generation tests, provides strong evidence that this system is secure from attack. Through the use of a dynamic key generation process, the complexity of attacking the system was raised to the point that to carry out an attack would have required an impractically high number of power traces (>100,000), and that no relationship to a key could have been observed in these traces, thereby rendering standard DPA attacks ineffective. Although the additional security provided by this design does come with the additional cost of increased logic resource usage (12.5% registers, 14.5% LUTs), I believe that for high assurance type applications, these costs are reasonable and in line with the overall low power consumption of ~0.5 mW from the system.

This research establishes an important benchmark. The promising results of this study pave the way for several compelling directions for future research:

- Architectural Optimization: Focus on reducing hardware overhead by exploring lighter-weight LSTM/RNN variants, implementing fixed-point arithmetic, and developing a dedicated 'neural accelerator' to replace general-purpose CPU cores.
- Expansion and New Standards: Extend the methodology to evaluate resistance against other side-channel attacks (e.g., Electromagnetic Analysis) and integrate it with additional cryptographic standards, including AES-192/256, ChaCha20, and post-quantum methods.
- Formal Security Verification: Quantify the security level guaranteed by LSTM randomization through concrete models and measurements, and consider using adversarial training that incorporates simulated power trace leakage metrics.
- System-level Integration: Develop efficient key synchronization protocols and integrate the cryptosystems into larger secure protocol stacks for practical deployment in environments like smart grids and Internet of Things (IoT) networks.

Acknowledgment

This research paper is not supported by any company or organization; it relies solely on the efforts of the authors.

Funding

Not applicable.

Conflict of Interest

The authors declare that they have no conflict of interest.

Data Availability

All data generated or analyzed during this study are included in this published article.

Author Contributions

All authors contributed to the study's conception and design. The specific contributions of each author are outlined below:
Khalil M. Abdelnaby: Conceptualization, Methodology, Software, Original Draft Preparation, Validation, And Project Administration.
Mohammed A. F. Al-Husainy: Formal Analysis, Investigation, And Supervision.
Mohammad O. Alhawarat: Resources, Data Curation, And Visualization.
Mohamed A. Rohaim: Writing – Original Draft Preparation.
Khairy M. Assar: Writing – Review, and Editing.

References

[1] Kumar, N., Mishra, V. M. & Kumar, A. Smart grid and nuclear power plant security by integrating cryptographic hardware chip. Nucl. Eng. Technol. 53, 3327–3334 (2021).
[2] Dubey, A., Karsai, G., Volgyesi, P., Metelko, M., Madari, I. & Tu, H. Device access abstractions for resilient information architecture platform for smart grid. *IEEE Embed. Syst. Lett.* 11, 34–37 (2019).
[3] Kumar, A., Vishnoi, P. & Shimi, S. L. Smart grid security with cryptographic chip integration. EAI Endorsed Trans. Energy Web 19, 1–12 (2019).
[4] Kumar, A., Bansal, K., Kumar, D., Devrari, A., Kumar, R. & Mani, P. FPGA application for wireless monitoring in power plant. *Nucl. Eng. Technol.* 53, 1167–1175 (2021).

[5] Metke, A. R. & Ekl, R. L. Security technology for smart grid networks. IEEE Trans. Smart Grid 1, 99–107 (2010).
[6] Katyara, S., Shah, M. A., Chowdhary, B. S., Akhtar, F. & Lashari, G. A. Monitoring, Control and Energy Management of Smart Grid System via WSN Technology Through SCADA Applications. Wirel. Pers. Commun. 106, 1951–1968 (2019).
[7] Farooq, Q., Shan L. & Yuan, H. Y. New Approach Towards Cyber Security for Nuclear Power Control System. 2021 IEEE Kansas Power and Energy Conference (KPEC), Manhattan, KS, USA,1-6 (2021).
[8] Kumar, N., Mishra, V. M. & Kumar, A. Smart grid security with AES hardware chip. Int. J. Inf. Technol. 12, 49–55 (2020).
[9] Kimani, K., Oduol, V. & Langat, K. Cyber security challenges for IoT-based smart grid networks. Int. J. Crit. Infrastruct. Prot. 25, 36–49 (2019).
[10] Mohamed, A. E., Jae, C. J. Development of field programmable gate arrayebased encryption module to mitigate man-in-the-middle attack for nuclear power plant data communication network. Nuclear Engineering and Technology. 50, 780-787 (2018).
[11] K. Gaj and P. Chodowiec, "FPGA and ASIC implementations of AES," in *Cryptographic Engineering*, Ç. K. Koç, Ed. Boston, MA: Springer, 2009, ch. 10, pp. 235–294.
[12] Proulx, A., Chouinard, J. Y., Fortier, P. & Miled, A. A Survey on FPGA Cybersecurity Design Strategies. ACM Trans. Reconfigurable Technol. Syst. 16, (2023).
[13] Magyari, A. & Chen, Y. Review of State-of-the-Art FPGA Applications in IoT Networks. Sensors 22, (2022).
[14] Guruprasad, S. P. & Chandrasekar, B. S. An evaluation framework for security algorithms performance realization on FPGA. 2018 IEEE Int. Conf. Curr. Trends Adv. Comput. ICCTAC 2018 1–6 (2018).
[15] Bentoutou, Y., Bensikaddour, E. H., Taleb, N. & Bounoua, N. An improved image encryption algorithm for satellite applications. Adv. Sp. Res. 66, 176–192 (2020).
[16] Xu, X. & Chen, S. An Optical Image Encryption Method Using Hopfield Neural Network. Entropy 24, (2022).
[17] Sarkar, A. A symmetric neural cryptographic key generation scheme for Iot security. Appl. Intell. 53, 9344–9367 (2023).
[18] Musanna, F., Dangwal, D. & Kumar, S. Novel image encryption algorithm using fractional chaos and cellular neural network. J. Ambient Intell. Humaniz. Comput. 13, 2205–2226 (2022).
[19] Mahalingam, H., Meikandan, P. V., Thenmozhi, K., Moria, K. M., Lakshmi, C., Chidambaram, N. & Amirtharajan, R. Neural attractor-based adaptive key generator with DNA-coded security and privacy framework for multimedia data in cloud environments. *Mathematics* **11**, 1769 (2023).
[20] Alzubi, J. A., Alzubi, O. A., Beseiso, M., Budati, A. K. & Shankar, K. Optimal multiple key-based homomorphic encryption with deep neural networks to secure medical data transmission and diagnosis. Expert Syst. 39, 1–17 (2022).
[21] Lin, Z., Wang, L., Cai, Y., Wang, F. & Wu, Y. Implementation of a built-in self-test for nuclear power plant FPGA-based safety-critical control systems. Ann. Nucl. Energy 165, 108644 (2022).
[22] PWRYichun, W., Xuanxuan, S., Yuanfeng C., Junyi Z., Zhiqian, W., & Jianxiang, Z. Development, verification and validation of an FPGA-based core heat removal protection system for a PWR. Nuclear Engineering and Design 301, 311–319 (2016).
[23] He, D., He, J., Liu, J., Yang, J., Yan, Q. & Yang, Y. An FPGA-based LSTM acceleration engine for deep learning frameworks. *Electronics* **10**, 681 (2021).
[24] Negabi, I., El Asri, S. A., El Adib, S. & Raissouni, N. Convolutional neural network based key generation for security of data through encryption with advanced encryption standard. Int. J. Electr. Comput. Eng. 13, 2589–2599 (2023).
[25] Lee, S. & Huh, J. H. An effective security measures for nuclear power plant using big data analysis approach. J. Supercomput. 75, 4267–4294 (2019).
[26] Xiong, W., Bai, T., Gu, P. F., Liang, H. H. & Tang, J. Z. Research on Static Testing Technology of Nuclear Safety-Critical Software Based on FPGA Technology. Lecture Notes in Electrical Engineering vol. 595 LNEE (Springer Singapore, 2020).
[27] Kim, J., Kim, E. S., Yoo, J., Lee, Y. J. & Choi, J. G. An Integrated Software Testing Framework for FPGA-Based Controllers in Nuclear Power Plants. Nucl. Eng. Technol. 48, 470–481 (2016).
[28] Iyer, S. Cyber security for smart grid, cryptography, and privacy. Int. J. Digit. Multimed. Broadcast. (2011).
[29] Rao, M. R. & Sharma, R. K. FPGA implementation of combined AES-128. 8th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2017 3–8 (2017).
[30] Hassen, M., Fatma, K., Belgacem, B., Mohsen, M. Ahigh-speed AES design resistant to fault injection attacks. Microprocessors and Microsystems 41, 47–55 (2016).
[31] Yang, S., Shim, H., Yang, W. & Kyung, C. M. A new RTL debugging methodology in FPGA-based verification platform. Proc. 2004 IEEE Asia-Pacific Conf. Adv. Syst. Integr. Circuits 180–183 (2004).
[32] Anandakumar, N. N., Hashmi, M. S. & Tehranipoor, M. FPGA-based Physical Unclonable Functions: A comprehensive overview of theory and architectures. Integration 81, 175–194 (2021).
[33] Chenglu, J., Vasudenv, G., Rameh, K., & Jeyavijayan, R. Security of Cloud FPGAs: A Survey. ACM Comput. Surv. (2020).
[34] Benhani, E. M., Aubert, A. & Bossuet, L. On the Security Evaluation of the ARM TrustZone Extension in a Heterogeneous SoC. 108–113 (2017).

[35] Trimberger, S., Mcneil, S., Fpgas, B. & Engines, C. Security ofFPGAs in Data Centers. 117–122 (2017).
[36] Zhang, J. & Qu, G. Recent Attacks and Defenses on FPGA-based Systems. 12, (2019).
[37] Ding, Z., Wang, Y., & Zou, X. The methods of FPGA software verification. 2011 IEEE International Conference on Computer Science and Automation Engineering, Shanghai, China, 2011.
[38] Nagaraj, R. Simulation and Hardware Implementation of FPGA Based Controller for Hybrid Power System. 3, 86–93 (2015).
[39] Kouser, Z., Manish, S., & Amit. M. J. FPGA Implementation of Advanced Encryption Standard Algorithm. IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2016), December 23-25, 2016.
[40] Nyangaresi, V. O. & Ahmad, M. Artificial neural network and symmetric key cryptography based verification protocol for 5G enabled Internet of Things. 1–22 (2022) doi:10.1111/exsy.13126.
[41] Wang, S., Hong, L. & Jiang, J. Optik An image encryption scheme using a chaotic neural network and a network with multistable hyperchaos. Optik (Stuttg). 268, 169758 (2022).
[42] Liang, B. F-LSTM : FPGA-Based Heterogeneous Computing Framework for Deploying LSTM-Based Algorithms. (2023).
[43] Guy, M., Xiaoming, Z., & Zhendong ,W. An FPGA Implementation of Stochastic Computing-based LSTM. 2019 IEEE 37th International Conference on Computer Design (ICCD). 38–46 (2019)
[44] Nguyen, X. & Pham-quoc, C. An FPGA-based Convolution IP Core for Deep Neural Networks Acceleration. 12, 1–8 (2022).
[45] Mohamed, N. & Cavallaro, J. Design and Implementation of an FPGA - Based DNN Architecture for Real - time Outlier Detection. J. Signal Process. Syst. (2023).
[46] Guan, Y., Yuan, Z., Sun, G. & Cong, J. FPGA-based Accelerator for Long Short-Term Memory Recurrent Neural Networks, 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), 2017.
[47] Hobden, P., Srivastava, S. & Nurellari, E. FPGA-Based CNN for Real-Time UAV Tracking and Detection. 3, 1–14 (2022).
[48] Wang, H. & Dubrova, E. Tandem Deep Learning Side - Channel Attack on FPGA Implementation of AES. SN Comput. Sci. 2, 1–12 (2021).
[49] Lee, S. & Huh, J. H. An effective security measures for nuclear power plant using big data analysis approach. J. Supercomput. 75, 4267–4294 (2019).
[50] Pour, M. M., Anzalchi, A. & Sarwat, A. A Review on Cyber Security Issues and Mitigation Methods in Smart Grid Systems. SoutheastCon 2017, Concord, NC, USA. 4, 1-41 (2017).
[51] J. Malik, A. Akhundzada, A. Al-Shamayleh, S. Zeadally, & A. Almogren, "Hybrid Deep Learning Based Threat Intelligence Framework for Industrial IoT Systems," J. Ind. Inf. Integr., vol. 45, 2025, no. 100846.
[52] K. Tiri & I. Verbauwhede, “A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation,” *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, vol. 1, pp. 246–251, 2004, doi: 10.1109/DATE.2004.1268852.
[53] S. Mangard, E. Oswald, & T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. New York, NY, USA: Springer, 2007.
[54] A. Moradi, A. Poschmann, S. Ling, C. Paar, & H. Wang, “Pushing the limits: A very compact and a threshold implementation of AES,” Advances in Cryptology – EUROCRYPT, pp. 69–88, 2011.
[55] C. Paar & J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin, Germany: Springer, 2010.
[56] D. Dolev & A. C. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
[57] L. Lamport, “Password authentication with insecure communication,” Communications of the ACM, vol. 24, no. 11, pp. 770–772, 1981.
[58] Y. Zhou, H. Ren, & Q. Zhang, “Deep learning-based side-channel attack and defense: A survey,” *IEEE Access*, vol. 9, pp. 100 982–101 001, 2021.
[59] J. Kim, Y. Shin, & D. Han, “Deep learning-based hardware security: Opportunities and challenges,” *IEEE Design & Test*, vol. 37, no. 5, pp. 30–40, 2020.
[60] M. Zaid, M. A. Hanif, & M. Shafique, “A survey on machine learning for side-channel analysis and countermeasures,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 16, no. 2, pp. 1–28, 2020, doi: 10.1145/3378587.
[61] Hiari, M., Alraba'nah, Y., & Qaddara, I.. A deep learning-based intrusion detection system using refined LSTM for DoS attack detection. Engineering, Technology and Applied Science Research, 15, (4), 25627–25633 (2025).
[62] Mosleh M. Abualhaj, Sumaya N. Al-Khatib, Mahran Al-Zyoud, Iyas Qaddara, Mohammad O. Hiari, and Sultan Mesfer A. Aldossary, “Enhanced Network Communication Security Through Hybrid Dragonfly-Bat Feature Selection for Intrusion Detection," Journal of Communications, vol. 20, no. 5, pp. 607-618, (2025).
[63] Almomani, O., Alsaaidah, A., Abu-Shareha, A. A., Alzaqebah, A., Almaiah, M. A. & Shambour, Q. Enhance URL Defacement Attack Detection Using Particle Swarm Optimization and Machine Learning. J. Comput. Cogn. Eng. 4, 296–308 (2025).
[64] Almaiah, M. A., Saqr, L. M., Al-Rawwash, L. A., Altellawi, L. A., Al-Ali, R. & Almomani, O. Classification of Cybersecurity Threats, Vulnerabilities and Countermeasures in Database Systems. Comput. Mater. Contin. 81, 1509–1528 (2024).
[65] A. Nasr, K. Mohamed, A. Elshenawy, and M. Zaki, "A Siamese deep learning framework for efficient hardware

Trojan detection using power side-channel data," *Sci. Rep.*, vol. 14, no. 1, p. 13013, 2024.
[66] B. M. Elomda, T. A. A. Abdelbary, H. A. Hassan, K. S. Hamza, and Q. Kharma, "An Enhanced Multi-Layer Blockchain Security Model for Improved Latency and Scalability," *Inf.*, vol. 16, no. 3, 2025.
[67] O. AbuAlghanam, H. Alazzam, M. Qatawneh, M. Alazab, and M. A. Alsharaiah, "Elevating Mobile Security: An Ensemble Approach for Enhanced Malware Detection with M-iForest," *Int. J. Interact. Mob. Technol.*, vol. 19, no. 10, pp. 129–151, 2025.
[68] R. Sardar, T. Anees, A. Al-Shamayleh, E. Mehmood, W. Khalil, A. Akhundzada, and F. S. Shaikh, "Challenges in Detecting Security Threats in WoT: A Systematic Literature Review," *Artif. Intell. Rev.*, vol. 58, no. 7, 2025, doi: 10.1007/s10462-025-11176-z.
[69] M. N. Khan, I. Ullah, A. K. Bashir, M. A. Al-Khasawneh, A. Arishi, N. S. Alghamdi, and S. Lee, "Reinforcement Learning-Based Dynamic Power Management for Energy Optimization in IoT-Enabled Consumer Electronics," *IEEE Trans. Consum. Electron.*, vol. 71, no. 3, pp. 8103–8114, Aug. 2025.
[70] E. B. Agyekum, B. Tarawneh, F. L. Rashid, P. Kumar, F. Odoi-Yorke, H. Togun, W. F. Fendzi Mbasso, and V. I. Velkin, "A systematic review of nuclear energy and public acceptance—A detailed analysis of evolution, emerging trends, and future research potentials," *Energy Explor. Exploit.*, vol. 43, no. 5, pp. 2276–2302, 2025.



Khalil M. Abdelnaby is an associate professor of Data Science, Artificial Intelligence Department, Faculty of Information Technology, Al-Ahliyya Amman University at Amman, Jordan. He received his M.Sc. and Ph.D. degrees in systems and computers engineering Dept. from Al-Azhar University in 2015 and 2019, respectively. His fields of interest include Artificial Intelligence, Smart Systems, Deep Learning, Data Mining, Mathematics, Robotics, Automotive Control Systems, Motion and Navigation Control, LLM Acceleration with FPGA, and Embedded systems for IoT signal processing.
Email: k.abdelnaby@ammanu.edu.jo



Mohammed A. F. Al-Husainy is a professor of the Networks and Cybersecurity Department, Faculty of Information Technology, Al-Ahliyya Amman University at Amman, Jordan. He received his B.Sc. degree in 1994 from the University of Mosul, Mosul, Iraq, then his M.Sc. degree in 1996 from Saddam University, Baghdad, Iraq, and his Ph.D. degree in 2002 from the University of Technology, Baghdad, Iraq. He has more than 20 years of experience in teaching most university courses in the Computer Science and IT field. His research interests are in the broad field of algorithm design, including multimedia data processing, scheduling algorithms, cybersecurity, cryptography, steganography algorithms, and user authentication.
Email: m.alhusainy@ammanu.edu.jo



Mohammad O. Alhawarat (Member, IEEE) received the Ph.D. degree in computer science from Oxford Brookes University, Oxford, U.K., in 2007. He is currently an Associate Professor with the Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman, Jordan. His research interests include Arabic natural language processing, machine learning, deep learning, and neural networks. Dr. Alhawarat is a professional member of ACM.
Email: m.hawarat@ammanu.edu.jo



Mohamed A. Rohaim is an Assistance Professor of Data Science, Artificial Intelligence Department, Faculty of Information Technology, Al-Ahliyya Amman University at Amman, Jordan. He received his M.Sc. and Ph.D. degrees in systems and computers engineering Dept. from Al-Azhar University in 2010 and 2015, respectively. His fields of interest include Artificial Intelligence, Deep Learning, Data Mining and Arabic natural language processing.
Email: m.rohaim@ammanu.edu.jo



Khairy M. Assar is an Ass. Professor at the Faculty of Engineering, Al-Azhar University, Cairo, Egypt. He received his B.Sc. in Computer and Systems Engineering from Al-Azhar University in 1997, his M.Sc. in Computer Engineering from Helwan University in 2004, and his Ph.D. from Al-Azhar University in 2010. His fields of interest include the Internet of Things (IoT), Smart City Systems, Light Communications (LiFi), Robotics, Industrial Automation, Optimization and Robust Control, and Computational Intelligence.
Email: khairyassar@azhar.edu.eg