electronics

MDPI

Review

Application of FPGA Devices in Network Security: A Survey

Abdulmunem A. Abdulsamad * and Sándor R. Répás

Department of Telecommunications, Széchenyi Istvan University, 9026 Győr, Hungary; repas.sandor@sze.hu
* Correspondence: abdulsamad.abdulmunem@sze.hu

Abstract

Field-Programmable Gate Arrays (FPGAs) are increasingly shaping the future of network security, thanks to their flexibility, parallel processing capabilities, and energy efficiency. In this survey, we examine 50 peer-reviewed studies published between 2020 and 2025, selected from an initial pool of 210 articles based on relevance, hardware implementation, and the presence of empirical performance data. These studies encompass a broad range of topics, including cryptographic acceleration, intrusion detection and prevention systems (IDS/IPS), hardware firewalls, and emerging strategies that incorporate artificial intelligence (AI) and post-quantum cryptography (PQC). Our review focuses on five major application areas: cryptographic acceleration, intrusion detection and prevention systems (IDS/IPS), hardware firewalls, and emerging strategies involving artificial intelligence (AI) and post-quantum cryptography (PQC). We propose a structured taxonomy that organises the field by technical domain and challenge, and compare solutions in terms of scalability, resource usage, and real-world performance. Beyond summarising current advances, we explore ongoing limitations—such as hardware constraints, integration complexity, and the lack of standard benchmarking. We also outline future research directions, including low-power cryptographic designs, FPGA–AI collaboration for detecting zero-day attacks, and efficient PQC implementations. This survey aims to offer both a clear overview of recent progress and a valuable roadmap for researchers and engineers working toward secure, high-performance FPGA-based systems.

Keywords: FPGA; network security; cryptographic acceleration; IDS/IPS; deep packet inspection; artificial intelligence; post-quantum cryptography; hardware firewalls; energy-efficient design; zero-day threat detection

check for updates

Academic Editor: Alexander Barkalov

Received: 7 September 2025
Revised: 29 September 2025
Accepted: 29 September 2025
Published: 30 September 2025

Citation: Abdulsamad, A.A.; Répás, S.R. Application of FPGA Devices in Network Security: A Survey. *Electronics* **2025**, *14*, 3894. https://doi.org/10.3390/electronics14193894

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As networks expand globally, the urgency of the cyber threat grows, with institutions and businesses increasingly exposed to Distributed Denial of Service (DDoS) attacks, phishing, and advanced persistent threats (APTs) [1]. The interconnectedness of digital infrastructures is increasing, expanding the attack surface and providing more opportunities for malicious actors [2]. Traditional software-based security systems, implemented on CPUs, are struggling to keep pace with the speed demands of modern networks due to their sequential processing limitations [1]. As a result, these systems often fail to prevent attacks that exploit fast-paced, high-throughput traffic conditions [3].

One solution to these issues can be found in the use of Field-Programmable Gate Arrays (FPGAs), which are flexible and reconfigurable [4]. Unlike dedicated hardware, these can be reprogrammed on the fly to respond to evolving attacks, making them particularly well-suited to cryptographic processing, threat measurement, and network traffic

analysis [1,5]. Moreover, the ability of FPGAs to execute multiple operations in parallel significantly enhances performance, enabling tasks such as encryption, packet inspection, and anomaly detection to run simultaneously with reduced latency [6]. The general architecture of an FPGA—consisting of Configurable Logic Blocks (CLBs), programmable interconnects, and input/output (I/O) interfaces—is shown in Figure 1 [6]. This structure illustrates the parallelism and reconfigurability that underpin their effectiveness in accelerating security-critical workloads.

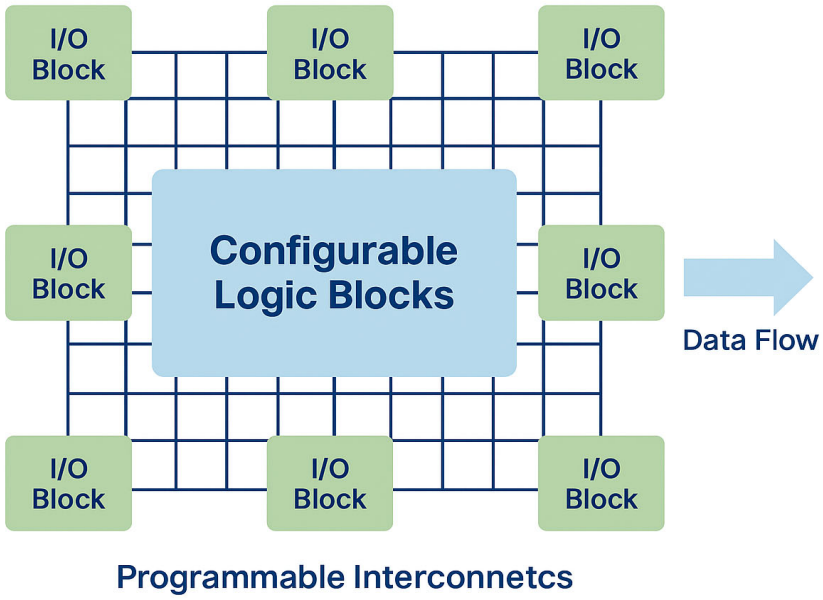

Figure 1. FPGA architecture [6].

Beyond architectural advantages, FPGAs consistently demonstrate measurable benefits over conventional CPU-based software solutions. As summarised in Table 1, FPGA-based designs offer higher throughput, lower latency, and improved energy efficiency. In contrast, software-based security systems are constrained by sequential execution and are more susceptible to performance bottlenecks. These distinctions underscore why FPGAs are increasingly regarded as a cornerstone for secure, high-performance network infrastructures.

Table 1. FPGA-Based Security V/S Traditional Software-Based Security.

Feature	FPGA-Based Security	Traditional Software-Based Security
Processing Speed	Possibility of high-speed parallel processing	Sequential, slower processing
Flexibility	Reconfigurable hardware logic	Software updates required
Power Efficiency	Reduced power usage for functions	Higher power requirements
Security Level	Secure at the hardware level, thus more challenging to compromise	More Susceptible to Software Exploits
Latency	Increased parallelism reduces latency	Sequential execution increases latency.
Scalability	Can effectively scale with hardware resources	Restricted by CPU cores and memory

Table 1 contrasts FPGA-based security solutions with traditional CPU-based software approaches. As shown, FPGAs offer significant advantages in terms of speed, parallelism, and energy efficiency, whereas CPU-based systems often encounter performance bottlenecks and higher power consumption. This comparison underlines why FPGAs are increasingly preferred for modern high-speed and real-time security applications.

FPGAs enhance threat detection, prevention, and mitigation across these applications, offering a versatile and high-performance security platform [7]. As cyberattacks continue to grow in scale and sophistication, organisations increasingly require hardware-assisted defences that move beyond reactive approaches. By integrating FPGA technology into security infrastructures, networks can achieve greater resilience, adaptability, and operational continuity in the face of evolving threats [1].

This paper explores how FPGAs improve network security in various areas, such as cryptographic acceleration, IDS/IPS, firewall implementations, and secure communication protocols (e.g., TLS/SSL and IPsec). FPGAs are increasingly being used in the field of cryptography to enhance encryption and decryption processes, ensuring more efficient and secure data transmission [8]. They enhance the efficiency of IDS/IPS by allowing real-time pattern matching and anomaly detection [1,3]. FPGA-based deep packet inspection (DPI) improves rule matching and malicious content filtering for firewalls [9].

Some other real-world applications of FPGA are mentioned as follows: FPGAs are widely used in secure hash functions, such as a hash function (MD5, SHA-2, SHA-3, etc.) that operates in the best circumstances to ensure that an input (or a message) is the same as a specific hash value, commonly used to maintain data integrity and authentication [10]. Array-based architecture features parallel processing, allowing hash functions to be calculated and optimised with lower power consumption and fewer cycle counts [11–15]. FPGAs also optimise algorithms for throughput using pipelining and loop unrolling, which are essential for real-time applications, such as blockchain verification or secure network authentication [16].

While several surveys have examined the use of FPGAs in general computing and embedded systems, only a limited number have provided a systematic review of their role in network security since 2020—a period characterised by the rapid convergence of high-speed networking, artificial intelligence (AI), and post-quantum cryptography (PQC). Earlier reviews tend to either concentrate narrowly on cryptographic primitives or offer broad overviews that lack benchmarking against modern requirements for performance, scalability, and energy efficiency. Critical aspects such as FPGA-based reconfigurability for mitigating zero-day attacks [1,17], energy-aware architectures [18], and the integration of FPGAs into AI-driven security pipelines [3,4] remain insufficiently explored.

This survey seeks to fill these gaps by

- Consolidating the most recent advances in FPGA-based defence mechanisms across cryptography, IDS/IPS, firewalls, and emerging paradigms;
- Introducing a structured taxonomy that organises these contributions by both domain and technical challenge;
- Identifying open research opportunities, including sustainable FPGA designs [18], FPGA–cloud co-integration, and hardware acceleration for post-quantum cryptographic standards [19].

In doing so, the paper offers more than a synthesis of existing findings: it provides actionable insights that bridge theoretical advances with deployable architectures. Unlike prior surveys that remain limited in scope or lack validation against real-world workloads, this review delivers a holistic, performance- and energy-aware analysis of FPGA applications across multiple layers of network security. By aligning academic progress with practical deployment constraints, the survey positions itself as both a comprehensive reference and a roadmap for guiding innovation in FPGA-based security research.

As illustrated in Figure 1, the general FPGA architecture is composed of Configurable Logic Blocks (CLBs), programmable interconnects, and input/output (I/O) interfaces. This reconfigurable structure supports parallelism, enabling FPGAs to accelerate cryptographic processing, intrusion detection, and secure communication [6].

2. Taxonomy of FPGA-Based Network Security Approaches

To provide a structured overview of the field, FPGA-based security research is grouped into five main categories:

1. Cryptographic Acceleration.
2. Intrusion Detection and Prevention Systems (IDS/IPS).
3. Network Defence Systems (Firewalls and VPNs).
4. Next-Generation FPGA Security (AI and Post-Quantum Cryptography).
5. Hardware Monitoring and Side-Channel Protection.

Figure 2 summarises the proposed taxonomy of FPGA-based security approaches. It categorises the literature into five functional areas—cryptographic acceleration, IDS/IPS, firewalls and VPNs, next-generation techniques, and hardware-level protection—while also highlighting key subfields such as SHA-3 acceleration, machine learning intrusion detection, and side-channel mitigation.

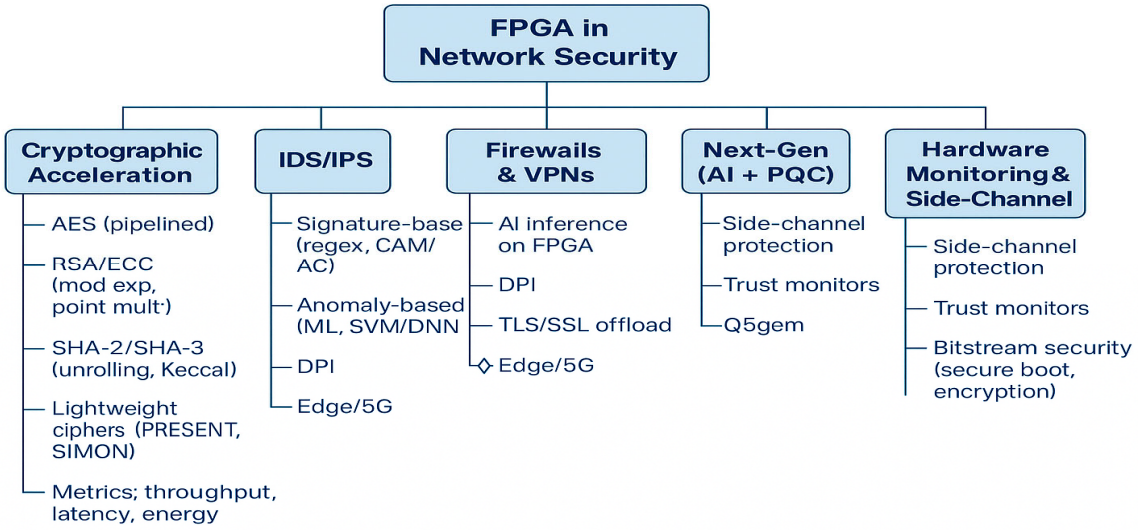

Figure 2. Taxonomy of FPGA-Based Network Security Approaches.

While these categories offer a structured framework, many FPGA-based systems span multiple domains. For instance, an IDS might use FPGA-accelerated cryptographic modules, or a firewall may incorporate AI-based threat detection.

To maintain clarity, each category in this taxonomy is defined by its **primary functional goal**:

- **Cryptographic Acceleration** focuses on core encryption, decryption, and hashing mechanisms such as AES, RSA, and SHA-3.
- **IDS/IPS** is centred on network traffic analysis and threat detection, including machine learning-enhanced anomaly detection.
- **Network Defence Systems** encompass firewalls and VPNs that manage secure traffic flow at the protocol level.
- **Next-Generation FPGA Security** addresses emerging paradigms like FPGA–AI co-design and post-quantum cryptography.
- **Hardware Monitoring and Side-Channel Protection** targets FPGA-level integrity, protecting against tampering, side-channel attacks, and unauthorised access.

To reduce redundancy, in-depth explanations are provided in the most relevant section, while other sections reference them as needed. For example,

- **SHA-3** is discussed in detail under **Cryptographic Acceleration**, with only brief mentions in IDS or blockchain contexts.

- **AI-based intrusion detection** is analysed in **IDS/IPS**, while broader AI integration frameworks are addressed in **Next-Generation Security**.

This taxonomy is not intended to impose strict divisions, but rather to provide a practical way to organise FPGA-based security research according to each area's primary focus. Because many systems span multiple roles—such as combining cryptographic functions with intrusion detection—we've structured the categories to highlight their core purposes. This approach helps reduce repetition and makes it easier for readers to follow how different FPGA technologies are being applied to today's cybersecurity challenges.

2.1. Cryptographic Acceleration

FPGAs have long played a vital role in accelerating encryption, decryption, and hashing operations. Thanks to their inherent parallelism and reconfigurability, they often outperform CPUs and GPUs in both speed and energy efficiency.

- Initial work focused on boosting SHA-2 performance for blockchain and IoT applications, achieving notable improvements in throughput and power efficiency [11,12].
- These efforts expanded to SHA-3, where optimised designs offer strong performance in constrained environments [15].
- Lightweight cyphers like PRESENT and SIMON were adapted to FPGAs for use in low-power networks such as wireless sensors [13,14].
- Hybrid solutions combining AES and ECC have been used to support both encryption and authentication in UAV communications [8].

These case studies demonstrate how FPGAs can support a wide range of cryptographic needs, from lightweight IoT encryption to high-speed hashing for secure data pipelines.

Table 2 summarises key cryptographic primitives implemented on FPGAs, including their security status, standardisation, hardware performance, and known limitations.

Table 2. Summary of Cryptographic Primitives Used in FPGA-Based Acceleration.

Cryptographic Primitive	Security Status	Standardization	FPGA Performance	Notes
SHA-2	Secure	NIST (FIPS 180-4)	3.2 Gbps @ 2.5 W on Zynq-7000 [12]	Deep pipelining improves throughput
SHA-3	Secure	NIST (FIPS 202)	2.1 Gbps @ 1.8 W on Artix-7 [13]	Optimised for low-area designs
PRESENT	Secure (lightweight)	ISO/IEC 29192-2	~1.8 Gbps on Spartan-6 [14]	Suited for low-power networks
SIMON	Some weaknesses	Unofficial (NSA)	~1.5 Gbps on Artix-7 [14]	Compact but weaker than PRESENT
AES + ECC Hybrid	Secure	NIST/ANSI X9	Platform-dependent [8]	Combines symmetric and asymmetric functions
MD5	Broken	Deprecated (RFC 6151)	High throughput (unspecified)	Do not use in adversarial contexts

Collectively, these examples highlight how FPGAs enable a range of cryptographic capabilities—from ultra-lightweight primitives to hybrid schemes—while striking a balance between energy efficiency, flexibility, and performance in secure system design.

2.2. Intrusion Detection and Prevention (IDS/IPS)

This category encompasses systems that monitor traffic in real-time and flag any suspicious activity. Since traditional CPU-based approaches often struggle to keep up in

high-throughput environments, FPGAs provide a means to maintain detection accuracy while handling fast-moving data streams.

- Early work focused on accelerating pattern matching in IDS tools like Snort, significantly reducing delays in identifying threats [1,20].
- More advanced designs now incorporate machine learning classifiers directly onto the FPGA, allowing for better recognition of unknown attacks, such as zero-days [3,17].
- FPGA-based IDS solutions have also found a place at the network edge—in 5G and IoT scenarios—where the ability to update detection rules on the fly helps systems adapt to shifting traffic behaviour [4].

To better understand the design trade-offs in AI-enhanced IDS, Table 3 compares representative studies in terms of datasets, evaluation metrics, model footprint, and handling of reconfiguration and concept drift. This provides insight into how such systems perform in long-running, real-world deployments.

Table 3. Summary of FPGA-Based AI-IDS Approaches.

Paper/Ref	Dataset Used	Eval. Metrics	FPGA Model Footprint (LUT/BRAM/DSP)	Reconfig. Overhead	Concept Drift Handling
Todorov et al. [17]	UNSW-NB15	Accuracy, F1-score	NR	Not discussed	None
Le Jeune et al. [18]	NSL-KDD, CIC-IDS2017	Accuracy, TPR, FPR	Reported: Moderate (NR exact)	Minimal (retrain offline)	Periodic retraining
Pham-Quoc et al. [1]	Custom synthetic + CIC	Accuracy, ROC-AUC	~38k LUTs, 64 BRAMs	Moderate (static config)	Manual model swap
Foudhaili et al. [3]	CIC-IDS2017	Accuracy, Latency	Lightweight: 12k LUTs, 40 DSPs	Low	Not addressed
Lin et al. [21]	NSL-KDD, TON_IoT	Accuracy, F1-score	28k LUTs, 55 BRAMs, 30 DSPs	Not reported	Adaptive (partial reconfig)

As the table shows, most FPGA-based AI-IDS report strong classification accuracy, but relatively few explicitly address reconfiguration overhead or long-term adaptability. These factors are crucial for real-world deployments, particularly in dynamic network environments.

2.3. Network Defence Mechanisms (Firewalls and VPNs)

In contrast to IDS/IPS that focus on identifying threats, this category includes mechanisms that enforce access control and ensure secure data transmission. FPGA-based designs are increasingly employed to implement high-performance firewalls, deep packet inspection (DPI) engines, and secure VPN tunnels.

Several studies demonstrate how FPGA-accelerated firewalls leverage parallel rule-matching architectures to maintain low-latency performance, even under heavy traffic loads [9,22]. In VPN scenarios, offloading cryptographic functions—such as those used in IPsec or WireGuard protocols—onto FPGA platforms has proven effective in reducing CPU load and improving gateway throughput [23].

A more nuanced application involves inspecting encrypted traffic. While traditional software tools often struggle with the real-time analysis of secure data flows, FPGAs have been used to extract and analyse flow-level features, such as packet length, timing patterns, and frequency. In some cases, such as the implementation described by Zhang et al. [24]. Session keys were preloaded or intercepted via a TLS proxy, enabling full decryption and DPI in hardware. These designs typically accelerate the data path while leaving session key negotiation and control-plane logic to software, a boundary often described as "pure hardware" inspection.

Ethical and Legal Considerations:

Encrypted traffic inspection raises essential ethical and legal considerations, particularly regarding user privacy, consent, and data sovereignty. Any system that decrypts or analyses protected traffic should only be deployed in environments with proper legal authorisation, such as enterprise-controlled networks or approved research testbeds. Developers and researchers should remain transparent about these capabilities and ensure compliance with relevant regulations and ethical standards.

2.4. Emerging Technologies: AI and Post-Quantum Cryptography (PQC)

As cybersecurity threats continue to evolve, there is a growing need to combine traditional hardware acceleration with forward-looking technologies. FPGAs are increasingly being used in conjunction with artificial intelligence (AI) models and post-quantum cryptographic (PQC) schemes to meet these emerging demands.

- In the AI space, FPGAs are proving especially useful in latency-sensitive environments like 5G edge computing. Real-time inference engines can be deployed directly on FPGA fabric, enabling adaptive packet filtering, behavioural anomaly detection, and real-time traffic classification. For instance, FPGA-accelerated convolutional neural networks (CNNs) and support vector machines (SVMs) have shown the ability to detect novel threats with minimal processing delay [4,25]. These setups combine software flexibility with the real-time processing power of dedicated hardware.
- PQC, designed to resist attacks from quantum computers, is another area where FPGAs are gaining traction. Hardware implementations of lattice-based algorithms like Kyber and NTRU have been successfully prototyped, demonstrating feasibility for real-time key exchange and encryption tasks in secure network channels [19]. While PQC schemes are more complex than classical cryptosystems, FPGAs allow developers to fine-tune resource allocation and execution pipelines, resulting in viable latency and power profiles.
- From a sustainability perspective, researchers have also explored techniques like Dynamic Voltage and Frequency Scaling (DVFS) to balance performance with energy savings [18]. These designs are beneficial in scenarios where the FPGA is expected to run 24/7 in high-availability environments.

The fusion of AI and PQC with FPGA hardware represents a key frontier in cybersecurity. These systems not only boost processing speed but also enable innovative, adaptable, and future-proof defence architectures.

2.5. Hardware Monitoring and Side-Channel Protection

While much of the focus in FPGA security is on accelerating cryptographic and network functions, another critical area is safeguarding the hardware itself. As attackers grow more sophisticated, vulnerabilities at the physical level—such as side-channel leaks, bitstream manipulation, and fault injection—have become increasingly relevant.

FPGAs are susceptible to side-channel attacks, where adversaries exploit power consumption, timing information, or electromagnetic emissions to infer secret data such as encryption keys. Notably, differential power analysis (DPA) and correlation power analysis (CPA) have been shown to break cryptographic implementations on FPGAs. These risks are exceptionally high in poorly randomised designs or those lacking signal balancing.

To address these threats, several countermeasures have been developed:

- **Power balancing techniques**, such as dual-rail logic and signal masking, aim to obscure correlations between power consumption and processed data.
- **Clock jittering** and dynamic frequency shifting introduce noise into the timing patterns, complicating time-based side-channel analysis.

- **Randomised execution paths help defend against fault injection and timing attacks by removing deterministic behaviour**.
- **Bitstream encryption and authentication** have been deployed to protect against tampering and reverse engineering of the configuration file, with devices supporting AES-based secure boot sequences.
- **Real-time watchdog monitors** embedded within the FPGA can track voltage, temperature, and current signatures to detect anomalies indicative of hardware tampering or malicious runtime behaviour.

One notable solution is **TrustGuard**, which uses signature-based analysis to identify suspicious power patterns at runtime, providing proactive defence against hardware-level exploits [26,27].

Table 4 provides a consolidated view of FPGA-based countermeasures against side-channel and hardware-level threats. Each defence is mapped to its threat model, example technique, expected trade-offs, and representative implementations from the literature.

Table 4. FPGA-based countermeasures for side-channel and hardware-level threats.

Countermeasure Type	Threat Model	Example Technique	Performance Trade-Off	Canonical Reference
Hiding (balancing)	Power analysis (DPA/CPA)	Dual-rail logic, balanced routing	Area ↑, Latency ↑	Zoni et al. [20]
Masking	Power analysis, glitch sensitivity	Threshold implementations, random shares	Design complexity ↑, Speed ↓	Jayasinghe [28]
Timing Obfuscation	Timing-based SCA, fault injection	Clock jitter, randomised delays	Latency ↑, Power noise ↑	Bommana et al. [29]
Bitstream Protection	Bitstream theft, reverse engineering	AES-encrypted config, secure boot	ROM use ↑, Secure flow req.	Meitei & Kumar [30]
Runtime Monitoring	Physical probing, fault attacks	TrustGuard (current/power analysis)	Resource overhead ↑, Alerts req.	Zhang et al. [27]
Access Control (e.g., JTAG)	Debug port misuse, backdoors	JTAG lockdown, scan disable, boot auth	Design setup ↑, Debug ↓	Rahman et al. [31]

As FPGA deployments expand into critical infrastructure and edge computing, integrating hardware-level defences is no longer optional. These protections must complement higher-layer security measures to ensure system-wide resilience—especially against sophisticated physical and side-channel threats.

Note on Methodological Transfer.

Some of the works referenced in this review—such as those on hyperspectral imaging or offline password recovery—are not directly applied to network security. However, they demonstrate transferable FPGA design methodologies (e.g., pipelining, lightweight hashing, side-channel protection) that have clear relevance to secure networking applications. These out-of-domain designs are summarised separately in Appendix A to maintain clarity while preserving valuable architectural insights.

2.6. Comparative Analysis of FPGA-Based Approaches

To move beyond a purely descriptive taxonomy, this section provides a comparative overview of selected FPGA-based implementations in cybersecurity. We focus on key performance indicators, including throughput, latency, power consumption, and energy efficiency. These metrics help reveal the strengths and trade-offs of different approaches across cryptographic acceleration, intrusion detection, and secure networking.

Table 5 summarises the results reported in the literature and includes energy efficiency, measured in gigabits per second per watt (Gb/s/W), whenever available. This metric is crucial for assessing the practicality of FPGA solutions in real-world deployments, particularly in power-constrained environments.

Table 5. Comparative Performance Metrics of FPGA-Based Network Security Designs.

Study	Application	FPGA Platform	Throughput (Gbps)	Latency (µs)	Power (W)	Energy Efficiency (Gb/s per W)	Notes
[12]	SHA-2	Zynq-7000	3.2	15	2.5	1.28	Deep pipelining applied
[13]	SHA-3	Artix-7	2.1	18	1.8	1.17	Optimised for low-area footprint
[9]	Firewall	Virtex-7	10	5	4.1	2.44	CAM-based rule matching
[17]	IDS (ML)	ZCU104	6	12	3.2	1.88	SVM classifier embedded
[23]	VPN (IPsec)	Kintex-7	5.5	7	3.7	1.49	Full hardware offload

As shown in Table 5, SHA-2 achieves the highest throughput at 3.2 Gbps [12], thanks to deep pipelining on the Zynq-7000 platform. SHA-3 trades off some speed for a smaller hardware footprint, making it more suitable for resource-constrained devices [13].

Firewall implementations, such as the design in [9], benefit from high parallelism and CAM-based architectures, achieving up to 10 Gbps throughput—though this comes at the cost of increased power consumption. On the other hand, machine-learning-driven IDS solutions [17] balance speed and intelligence by embedding classifiers, such as SVMs, directly into the FPGA fabric to detect anomalies with high accuracy. Their energy efficiency is respectable, but design complexity and latency must be considered.

VPN gateways using full offload architectures, as demonstrated by Liu et al. [23]. Achieve low latency while relieving the host CPU of cryptographic processing. Their performance is well-balanced for secure tunnelling applications in enterprise and industrial networks.

To aid cross-comparison, energy efficiency (Gbps per Watt) is reported where possible in Table 5. This helps assess the suitability of each design in performance- or power-constrained settings.

Figure 3 illustrates throughput versus power across selected FPGA security designs, highlighting the trade-offs in energy efficiency. It shows that while firewalls achieve the highest throughput, they also consume more power; in contrast, cryptographic primitives like SHA-3 offer moderate throughput with lower energy demands.

Figure 4 offers a latency comparison, illustrating the performance differences between basic cryptographic tasks and more complex IDS or VPN implementations. Lower-latency firewalls and VPNs can be critical for real-time packet processing, while ML-based IDS may trade latency for adaptability.

These visual summaries complement the tabular data by emphasising quantitative differences that might not be immediately apparent. They also serve as a reference point for selecting suitable FPGA designs based on application-specific performance and energy constraints.

To further contextualise the performance trade-offs discussed above, we introduce a layered architectural view of FPGA-based security solutions. Figure 5 visually maps each taxonomy category to its corresponding OSI layer(s), showing how FPGA functions—such as encryption, traffic inspection, or anomaly detection—are deployed across endpoints, edge devices, and gateways. This overview clarifies how specific data path locations align with different packet or flow features.

To complement this visual layout, Table 6 summarises the taxonomy-to-layer mapping in tabular form. It includes exemplar implementations from the literature, highlighting each design's primary function, OSI role, deployment context, and rationale for placement.

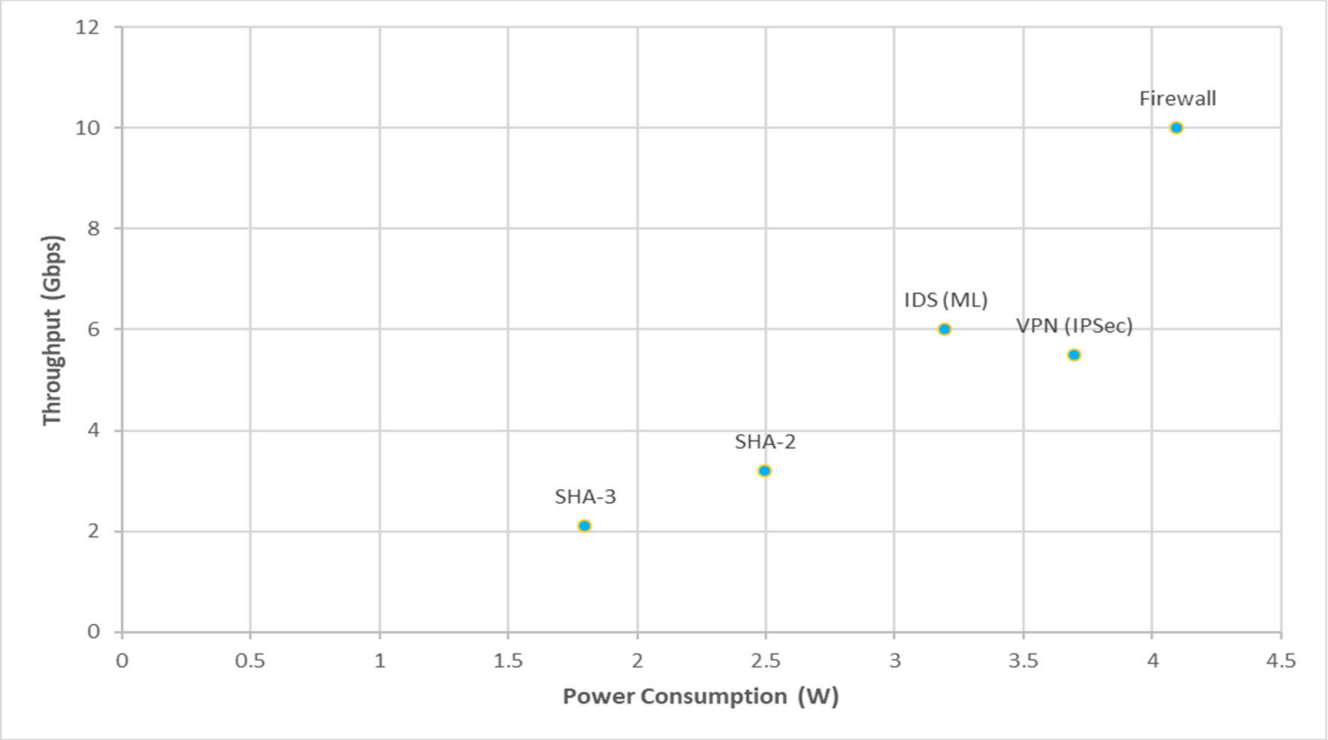

Figure 3. Throughput vs. Power Consumption of Selected FPGA-Based Security Designs.

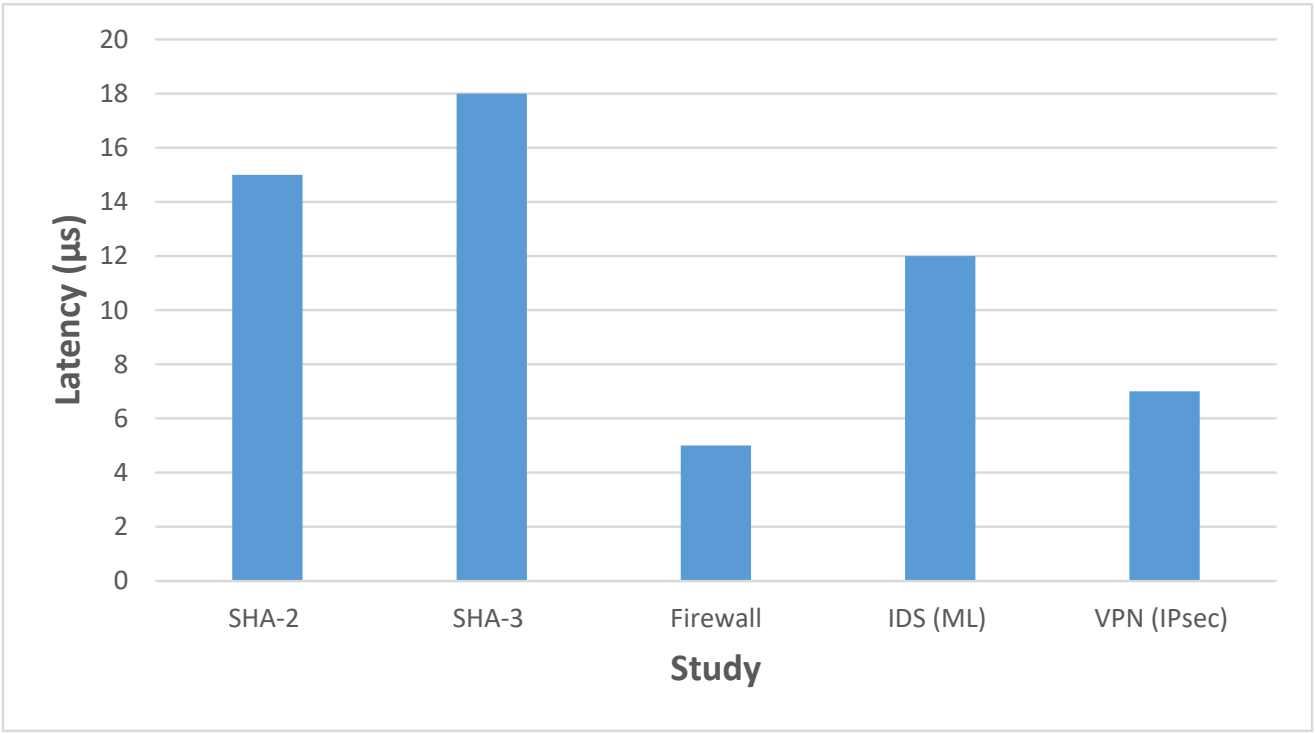

Figure 4. Latency Comparison Across FPGA-Based Security Applications.

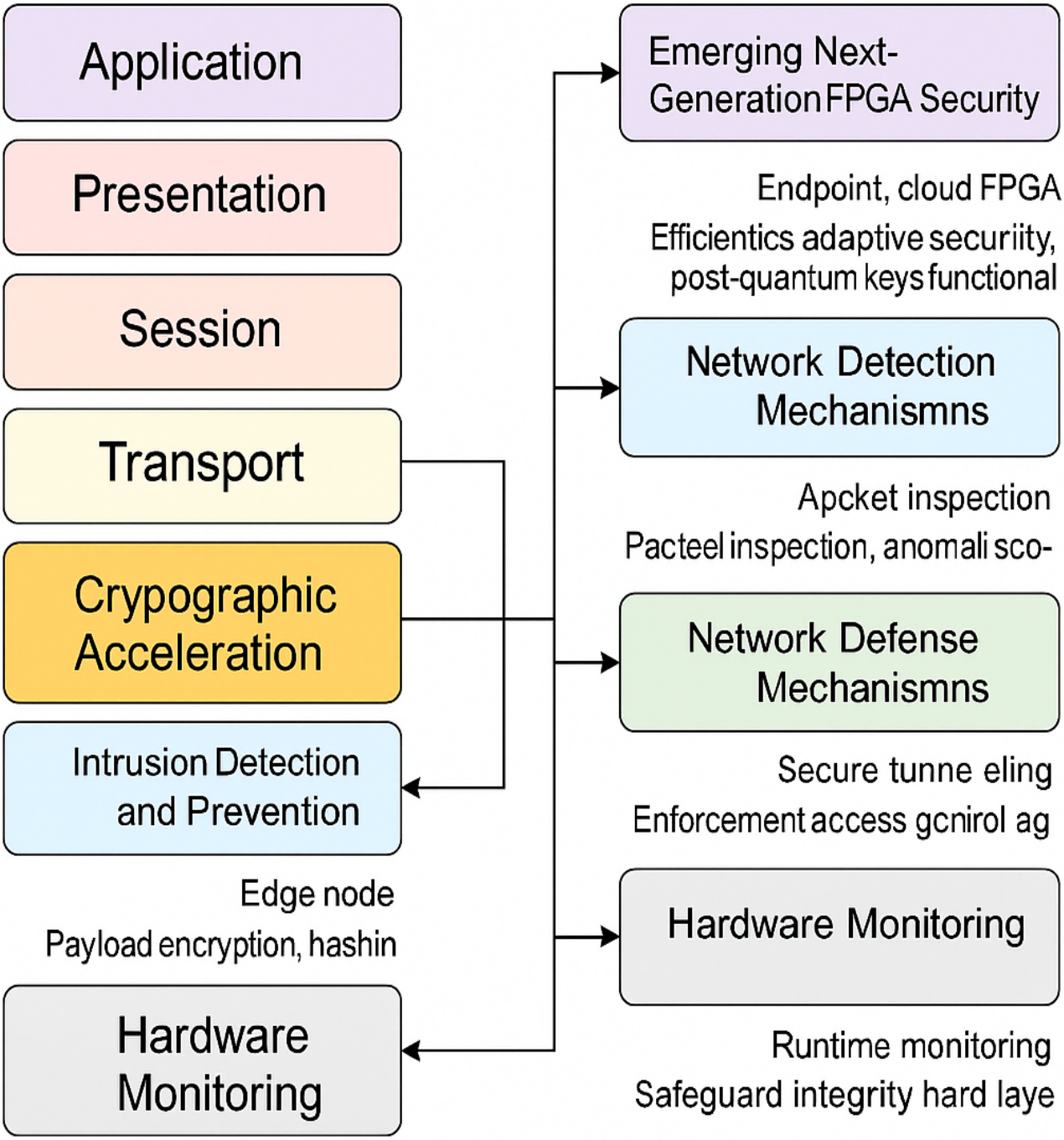

Figure 5. Taxonomy–OSI Layer Mapping.

Table 6. Mapping FPGA-Based Security Approaches to OSI Layers and Deployment Contexts.

Taxonomy Category	OSI Layer(s)	Device Placement	Data Path Location	Example Packet/Flow Features	Canonical Exemplar with Rationale
Cryptographic Acceleration	Layer 6–7 (Presentation/Application)	Endpoints, secure gateways	Payload encryption/decryption	Ciphertext blocks, hash digests	KAmmoun et al. [12] implemented an FPGA-based SHA-2 accelerator on the Zynq-7000, achieving high throughput through deep pipelining, demonstrating how FPGAs enhance payload security at the application layer.
IDS/IPS	Layer 3–4 (Network/Transport)	Edge routers, gateways	Packet inspection and anomaly detection	Headers, flow statistics, anomaly scores	Pham-Quoc et al. [1] designed an FPGA–AI IDS using the Zynq platform, enabling real-time anomaly detection and demonstrating the suitability of FPGA for fast, adaptive intrusion prevention.
Network Defence (Firewalls/VPNs)	Layer 3–4 (Network/Transport)	Perimeter firewalls, VPN gateways	Secure tunnelling, deep packet inspection	Source/destination IP, ports, encrypted payloads	Liu et al. [23] developed a WireGuard VPN gateway on an FPGA with low-latency offload, illustrating FPGA advantages for secure tunnelling and access control at the transport layer.
Next-Generation (AI & PQC)	Cross-layer (2–7)	Edge nodes, cloud accelerators	Adaptive decision-making, secure key exchange	Behavioural patterns, lattice-based keys	Duke-bergman & Huynh [19] prototyped FPGA-based PQC primitives for SPHINCS+, demonstrating the feasibility of quantum-resistant security in real-world FPGA deployments.
Hardware Monitoring & Side-Channel Protection	Layer 1–2 (Physical/Data Link)	FPGA devices, embedded systems	On-chip monitoring, runtime assurance	Power traces, timing signals, EM emissions	Zoni et al. [20] surveyed FPGA countermeasures against side-channel attacks, emphasising the importance of runtime monitoring in safeguarding physical-layer trust.

3. Methodology

This survey employs a structured, multi-phase review methodology designed to strike a balance between comprehensiveness, rigour, and reproducibility. The approach was inspired by systematic review practices such as the PRISMA 2020 framework [32], ensuring that the process is transparent and replicable.

3.1. Literature Search

A systematic search was conducted across leading scientific databases, including IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and MDPI Journals. The scope was limited to publications between January 2020 and January 2025.

Search queries combined keywords and Boolean operators to maximise coverage, for example,

- "FPGA" AND "network security".
- "FPGA" AND ("cryptography" OR "AES" OR "SHA").
- "FPGA" AND ("IDS" OR "IPS" OR "intrusion detection").
- "FPGA" AND ("firewall" OR "deep packet inspection").
- "FPGA" AND ("post-quantum cryptography" OR "AI").

This strategy yielded an initial dataset of 210 articles.

3.2. Screening and Selection

The screening process was conducted in two stages:
Title and abstract screening → removed 110 papers (duplicates, irrelevant works).
Full-text review → excluded 40 papers that
Focused solely on software-based implementations.
Lacked direct relevance to FPGA-based security.
Were not peer-reviewed or were not written in English.
This process resulted in **50 studies** being included in the final dataset.

3.3. Inclusion and Exclusion Criteria

Studies were included if they

- Directly addressed FPGA implementations in network security.
- Reported quantitative performance metrics (throughput, latency, energy efficiency, scalability).
- Introduced novel FPGA-based approaches (e.g., AI-enhanced IDS, post-quantum cryptography).
- Studies were excluded if they were:
- Redundant or purely theoretical.
- Unrelated to FPGA implementations.
- Non-security focused.
- Lacking empirical validation.

3.4. Categorization

Eligible studies were then **classified into a taxonomy** of five primary domains:

1. Cryptographic Acceleration.
2. Intrusion Detection and Prevention Systems (IDS/IPS).
3. Network Defence Systems (Firewalls and VPNs).
4. Next-Generation FPGA Security (AI and PQC).
5. Hardware Monitoring and Side-Channel Protection.

This classification provided a structured framework to compare methods, evaluate contributions, and identify limitations across the literature.

3.5. Data Extraction and Tabulation

For each study, detailed information was extracted across the following fields:

- **Device family** (e.g., Virtex, Zynq, Stratix).
- **Toolchain** (Vivado, Quartus, HLS frameworks).
- **Clock frequency** (MHz).
- **Resource utilisation** (LUTs, BRAMs, DSPs).
- **I/O width and traffic profile** (packet size, workload, dataset).
- **Performance metrics** (throughput in Gbps, latency in µs, energy efficiency in Gbps/W, scalability).

The extracted data were organised into Tables 5 and 6, which consolidate metrics across cryptographic, IDS/IPS, firewall, and password auditing domains. **Missing values were explicitly noted as "NR" (Not Reported).**

This structured tabulation ensures comparability across heterogeneous studies and highlights gaps in standardised reporting [32].

3.6. Visual Representation

The overall review process is summarised in Table 7 and illustrated through a PRISMA-style flow diagram (Figure 6), showing the progression from the initial 210 records to the final 50 included studies. This combination of tabular and visual representation reinforces the transparency and rigour of our methodology.

Table 7. Survey Methodology Summary.

Step	Description	Details
Database Search	Digital libraries queried	IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, MDPI Journals
Keywords	Main query terms	"FPGA", "Network Security", "Cryptography", "Intrusion Detection", "Firewall", "Post-Quantum", "Side-channel"
Time Frame	Years considered	2020–2025
Initial Results	Total papers found	210 articles
Screening (Title/Abstract)	Removed duplicates & unrelated works	110 removed
Full-Text Review	Applied relevance to FPGA + security	60 papers retained
Final Inclusion	Works included in the survey	50 selected
Exclusion Criteria	Not included	Pure software-only works, FPGA unrelated, non-security focus, non-English papers

As illustrated in Figure 6, the screening process progressively reduced the initial 210 records to the final set of 50 included studies. This flow diagram complements the structured summary in Table 3, reinforcing the clarity and transparency of the selection process.

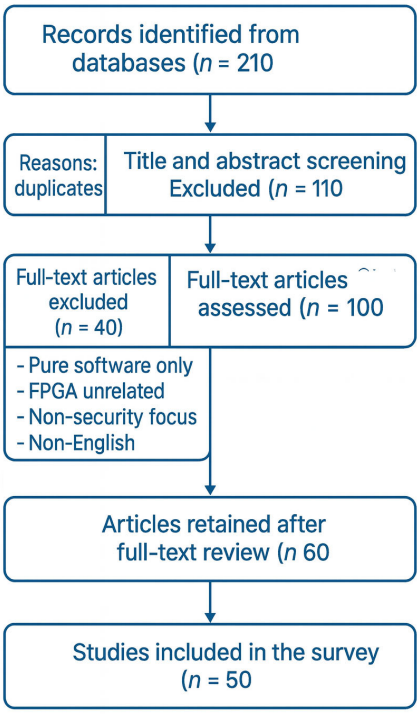

Figure 6. Literature Selection Flow (2020–2025).

4. Application of FPGA in Network Security

4.1. Cryptographic Applications of FPGA

Cryptographic algorithms are at the heart of secure communication channels, defined by data integrity, confidentiality, and authentication. Unlike traditional cryptographic solutions based on general-purpose CPUs, which provide high performance on a limited number of cores and are challenging to scale, FPGA-based implementations significantly improve throughput, reduce latency, and lower energy consumption [1,4].

4.1.1. Secure Hash Functions

Network security protocols often rely on cryptographic hash functions, such as MD5, SHA-2, and SHA-3, to ensure data integrity and authentication [1]. Traditional CPU-based implementations of these algorithms typically exhibit low throughput and high latency, which can lead to performance bottlenecks when migrating them directly to FPGA hardware [12]. By contrast, FPGA-based implementations can exploit parallelism, pipelining, and reconfigurability to achieve higher throughput and lower latency [33].

- **MD5 Accelerated:** MD5, though not safe, is still valuable for non-critical applications such as checksum computations and file integrity checking [33,34]. More recently, FPGA-based accelerators have provided improvements in MD5 processing through parallel hashing operations, yielding advantageous aspects for large-scale data validation and real-time integrity verification use cases [10]. Recent FPGA-based implementations of MD5 have focused on parallel hashing techniques, which are especially useful in scenarios that involve validating large volumes of data quickly, such as in real-time system integrity checks and large-scale storage verification [10].
- **SHA-2 and SHA-3 Optimisation:** SHA-2 (e.g., SHA-256, SHA-512) and SHA-3 (Keccak) offer stronger security and are widely used in encryption, digital signatures, and blockchain applications [11,13–15,25,33,35]. FPGA implementations enhance their performance through pipelining, parallel processing, and loop unrolling, optimising hashing speed and efficiency [12,13,36].

- **SHA-3 FPGA Implementation:** SHA-3 was designed to provide stronger security than its predecessors, and this strength is preserved when implemented on FPGAs. The challenges in hardware are not related to weakened cryptographic security, but rather to trade-offs in performance and resource usage. FPGAs, however, can handle SHA-3's memory and parallelism requirements effectively, enabling high-speed and energy-efficient cryptographic hashing that is well-suited for secure communications and blockchain applications [15,25,33]. Recent studies also highlight alternative primitives such as BLAKE3, whose pipelined FPGA implementations achieve competitive throughput and reduced power consumption [37]. The memory requirements of the SHA-3 algorithm can be managed efficiently in the hardware so that the FPGA-based SHA-3 implementations can offer high-speed cryptographic hashing at a lower power consumption cost, making them a valuable asset for blockchain networks and secure communications [15]. Many of these applications include financial services, healthcare, and blockchain technologies, where FPGA-based cryptographic hashing significantly reduces computational overhead while enabling real-time security provision [10–15,36]. In practice, these FPGA-accelerated hashing techniques secure mission-critical systems, ranging from medical devices to IoT gateways and blockchain networks, by ensuring real-time data integrity with minimal latency and power consumption.

4.1.2. Symmetric and Asymmetric Encryption

Data confidentiality and integrity in modern networks are ensured by encryption algorithms such as AES and RSA [19,20]. FPGA implementations can use parallelism and dedicated resources to achieve significant improvements in the speed of encryption and decryption operations [1,4].

- AES on FPGA:

The Advanced Encryption Standard (AES) is one of the most widely used symmetric encryption algorithms and remains a cornerstone of modern data protection. It secures applications ranging from online banking and cloud storage to everyday HTTPS communication. While AES is well known for its robustness and efficiency, software-only implementations often fall short when networks demand both high throughput and low latency. This limitation is particularly evident in large-scale systems such as Virtual Private Networks (VPNs), encrypted web traffic, and cloud services, where millions of encryption and decryption operations must be handled simultaneously [22].

Implementing AES on Field-Programmable Gate Arrays (FPGAs) addresses these challenges by leveraging their inherent parallelism and reconfigurability. In FPGA-based designs, the core AES functions—SubBytes, ShiftRows, MixColumns, and AddRoundKey—are distributed across pipelined hardware stages, allowing several operations to be executed concurrently. This approach significantly reduces processing time compared to sequential CPU or even GPU implementations [1]. Additionally, because the hardware can be tailored at the circuit level, FPGA-based AES designs are better positioned to resist side-channel attacks than purely software-based solutions.

The AES process on the FPGA begins with plaintext input and key expansion, followed by an initial AddRoundKey operation. The data then passes through $N-1$ rounds, each combining SubBytes (non-linear substitution using S-boxes), ShiftRows (row-wise transposition), MixColumns (matrix multiplication in Galois Fields), and AddRoundKey (XOR with the round key). In the final round, the MixColumns step is omitted in accordance with the AES specification. Figure 4 illustrates this architecture, showing how the algorithm maps naturally onto FPGA pipelines for efficient execution.

What makes this hardware approach particularly attractive is the way it leverages FPGA resources such as Look-Up Tables (LUTs), Digital Signal Processors (DSPs), and Block RAMs (BRAMs). These elements allow for customised datapaths, low-latency processing, and parallel handling of multiple encryption tasks. Pipelining means that while one block of data is being processed through SubBytes, another can move through ShiftRows or AddRoundKey, ensuring continuous throughput and efficient resource utilisation.

Beyond raw speed, FPGA-based AES implementations also deliver significant improvements in energy efficiency, which makes them highly suitable for embedded and real-time applications. Practical use cases include securing IoT devices, encrypting real-time video streams, and authenticating blockchain transactions. Furthermore, the reconfigurable nature of FPGAs enables designers to develop lightweight AES variants for resource-constrained systems or integrate AES with algorithms such as Elliptic Curve Cryptography (ECC) to provide multi-layered protection.

In summary, AES implementations on FPGAs offer a combination of performance, adaptability, and enhanced security. By exploiting parallelism, hardware-level optimisations, and reconfigurability, they provide a future-ready solution for secure communication, data storage, and embedded system protection [1,22].

The AES implementation in an FPGA is depicted in Figure 7. The figure illustrates how the AES core operations—SubBytes, ShiftRows, MixColumns, and AddRoundKey—are implemented in pipelined hardware stages to perform in parallel, thereby achieving higher throughput.

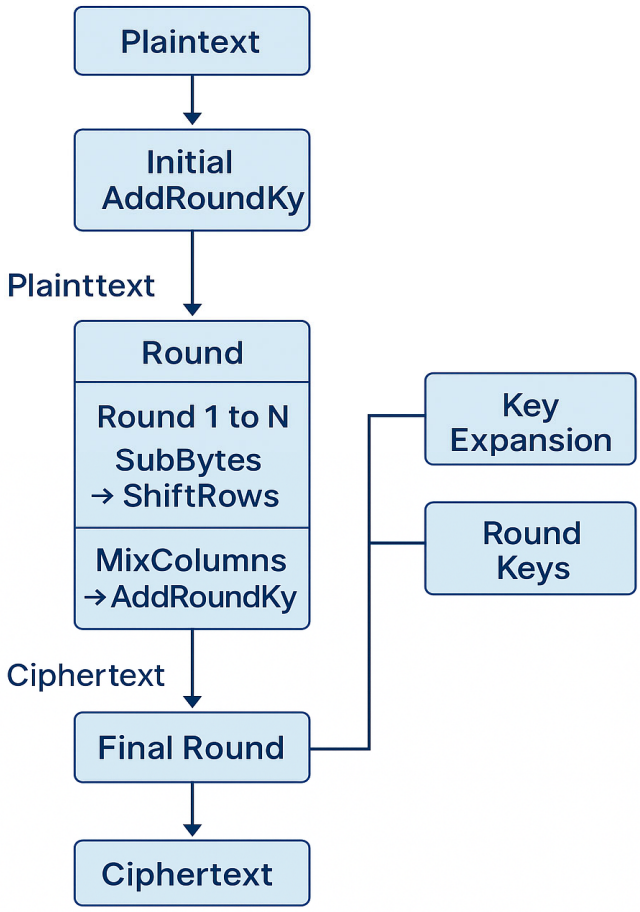

Figure 7. AES Encryption Architecture on FPGA.

As illustrated in Figure 7, this pipeline structure enables continuous data movement; therefore, several nodes are processed in parallel at any given stage. This architecture

significantly reduces latency and enhances scalability, making FPGA implementations of AES particularly suitable for high-throughput, real-time communication systems.

- RSA and ECC on FPGA:

Public-key cryptographic algorithms such as Rivest–Shamir–Adleman (RSA) and Elliptic Curve Cryptography (ECC) are fundamental to secure digital communication. They are widely used for key exchange, authentication, and digital signatures, but both involve computationally expensive mathematical operations. For RSA, the critical task is modular exponentiation, while ECC relies on elliptic curve point multiplication. When implemented purely in software, these operations can introduce significant latency, especially in real-time communication systems.

By mapping these functions onto Field-Programmable Gate Arrays (FPGAs), it becomes possible to achieve near real-time performance while also improving scalability and energy efficiency [8]. FPGA-based acceleration leverages parallelism and dedicated arithmetic resources to accelerate modular exponentiation in RSA and optimise elliptic curve point multiplication in ECC. These enhancements not only increase throughput but also improve the practicality of public-key cryptography in mobile security, blockchain applications, and large-scale authentication frameworks [23]. In this way, FPGA-based designs help address the growing demand for secure, low-latency cryptographic operations, positioning them as key enablers for protecting digital assets in next-generation networks [1].

Figure 8 illustrates the parallel processing flows of RSA and ECC on an FPGA. On the left, the RSA pipeline begins with input handling, passes through modular exponentiation in the RSA core, and produces the final output, with FPGA arithmetic units optimising the computations. On the right, the ECC pipeline starts with elliptic curve parameter initialisation, followed by key generation and point multiplication operations that underpin encryption and decryption. Together, these hardware-mapped datapaths demonstrate how FPGAs can execute RSA and ECC workloads simultaneously, offering flexible and high-performance solutions for secure digital communication.

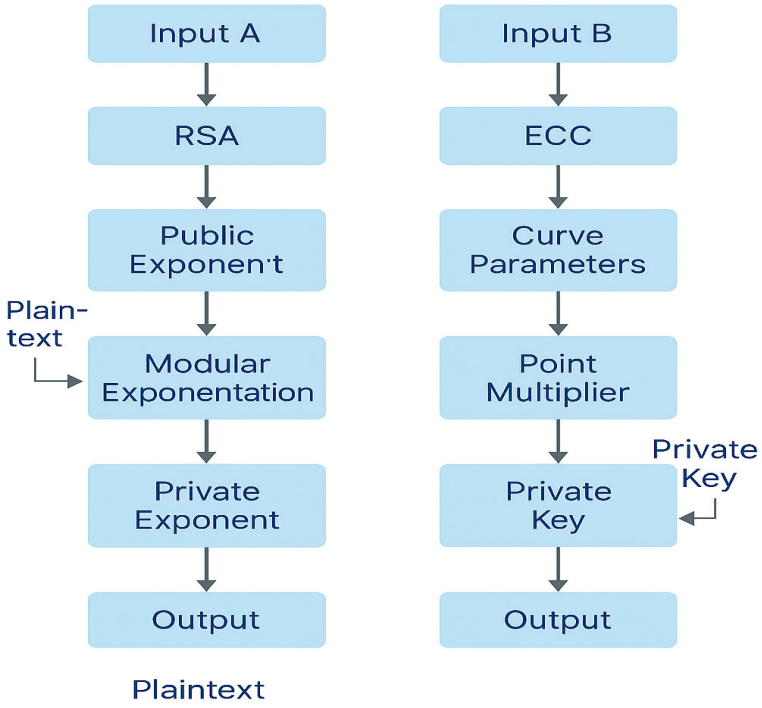

Figure 8. RSA and ECC Architectures on FPGA.

Building on the architectural insights illustrated in Figures 7 and 8, Table 8 offers a comparative overview of FPGA-based cryptographic algorithms. It outlines the key strengths of AES, RSA, and SHA-3 when mapped onto FPGA hardware and identifies their primary areas of application. This comparison underscores how FPGA acceleration can simultaneously enhance performance and strengthen security across a wide range of cryptographic workloads.

Table 8. FPGA-based cryptographic algorithms and their practical applications.

Algorithm	Advantages of an FPGA	Use Cases
AES	Resistance to side-channel attacks, High-speed encryption	VPNs, HTTPS, and cloud security
RSA	Optimised fast, modular exponentiation for key exchange	Secure messaging, digital signatures
SHA-3	Fast hashing, less power consumption	Data integrity, authentication, and Blockchain

While Table 8 summarises the main strengths and application domains of FPGA-based cryptographic algorithms, its broader implication is clear: FPGAs consistently provide a balance of speed, efficiency, and adaptability that software-only solutions struggle to match. For example, AES benefits from FPGA's pipelined execution to deliver high-throughput encryption suitable for real-time communications, whereas RSA acceleration enables practical large-scale key exchanges. Similarly, SHA-3 implementations achieve not only faster hashing but also lower power consumption, which is critical for blockchain and IoT environments. Together, these comparisons highlight how FPGAs can transform cryptographic workloads from potential performance bottlenecks into scalable, energy-conscious, and secure building blocks for modern digital infrastructures.

4.2. FPGA-Accelerated Password Auditing for Enhanced Security

4.2.1. The Role of Password Auditing in Cyber Defence

As passwords remain the leading method for authentication, password auditing is a critical component of any cybersecurity framework. Weak, reused, or poorly stored passwords are still one of the most exploited vulnerabilities by attackers using brute-force, dictionary, and credential-stuffing attacks [30]. Password auditing of this type is a proactive security measure in which stored credentials are systematically analysed, tested against known vulnerabilities, and robust authentication policies are developed to minimise the risk of compromise. The performance and scalability of traditional software-based auditing tools are often limited as threat actors become more sophisticated, and hashing mechanisms become more complex (e.g., sha512crypt and bcrypt) [31].

4.2.2. FPGAs as a Platform for High-Performance Password Cracking

Field-Programmable Gate Arrays (FPGAs) have emerged as a high-performance platform for accelerating password auditing tasks [38]. FPGA devices are reconfigurable hardware capable of executing thousands of operations in parallel, making them particularly well-suited to the repetitive and computationally intensive processes required for password recovery. For example, an FPGA implementation with 15 SHA-512crypt cores achieved a rate of approximately 675 password guesses per second, surpassing high-end CPUs such as the AMD Ryzen 9 and Apple M1 Max [30,39]. These findings demonstrate that FPGAs can deliver realistic performance for simulating attack scenarios while also providing defenders with a practical means to identify and remediate weaknesses in password hygiene before they are exploited.

4.2.3. Custom Cryptographic Modules and Real-Time Detection

FPGA hardware is reconfigurable, allowing for the design of custom modules to perform specific cryptographic operations [40]. *It has been* shown that FPGAs can utilise SHA-1 cores in WPA/WPA2 key recovery, decreasing the time and improving speed by about 40 times compared to the software version. Similarly, ref [38] highlighted the importance of field-programmable gate arrays (FPGAs) in the monitoring System-on-Chip (SoC) settings for password threat detection, enabling real-time identification through dynamic re-configuration. These capabilities demonstrate that password auditing has transitioned from a basic policy compliance verification check into a technically sophisticated and hardware-accelerated process.

4.2.4. Energy-Efficient Password Auditing with FPGAs

Besides raw computation bandwidth, any FPGA-based password auditing system can provide significant energy efficiency [30]. Brute-force or dictionary-based evaluations to exhaustively crack passwords on CPUs and GPUs are a costly endeavour in terms of energy, in addition to the time required for the attack to succeed. In contrast, FPGAs can match or exceed software-offload throughput while consuming lower wattage per workload, making them better suited for continuous or enterprise-wide audit operations. This becomes more significant for security operations centres (SOCs) and forensic labs, which require running long-term assessments while having a low operational cost.

4.2.5. Reconfigurability and Adaptation to Evolving Standards

Additionally, the reprogrammable nature of field-programmable gate arrays (FPGAs) plays a significant role in their applicability to the field of cybersecurity [38]. FPGA-based systems can also be re-synthesised to utilise updated cryptographic standards and new hashing algorithms as cryptography standards evolve, unlike ASICs, which are typically obsolete. Why such a flexible design? Long-term systems can adapt to new threats, and future-proof security is always a goal of industry leaders.

4.2.6. Applications in IoT and Embedded Security Systems

Perhaps one of the most poignant usages of FPGA-based password auditing is embedded in Internet of Things (IoT) settings [40]. These systems often employ lightweight and/or custom hashing algorithms that general-purpose processors do not support well. FPGA-auditing solutions can be tailored to target such unique implementations, thereby allowing for precise and efficient evaluation capabilities. Moreover, hybrid CPU-FPGA systems have demonstrated another possible way to leverage the elasticity of software and the performance of hardware, supporting real-time auditing and timely incident response in heterogeneous, renderer-strengthened environments [30].

4.2.7. Barriers and Challenges to Widespread Adoption

Despite this, FPGA-based solutions face barriers to adoption [38]. However, developing and implementing FPGA-targeted auditing frameworks involves expertise in digital hardware and knowledge of high-level synthesis tools. The high upfront cost of deploying an FPGA platform may also hinder smaller organisations. However, continuous research is being conducted to make FPGA coding easier and to create libraries of reusable, pre-optimised cryptographic cores, which can significantly lower the barrier to entry.

Beyond technical obstacles, ethical and regulatory considerations represent equally essential challenges. Password auditing tools—whether software or FPGA-accelerated—must be used under strict authorisation to prevent unauthorised access and misuse. International guidelines such as NIST's Technical Guide to Information Security Testing and Assessment

(SP 800-115) [41] and the ISO/IEC 27001 information security standard [29] explicitly highlight the need for prior approval, well-defined scope, and accountability when conducting password strength evaluations. Unauthorised use not only raises serious legal and ethical concerns but also exposes organisations to liability in the event of data misuse. Therefore, ensuring compliance with established ethical frameworks is as critical as advancing the hardware and algorithmic capabilities of FPGA-based auditing systems.

4.2.8. The Future of FPGA-Enhanced Password Security

The use of FPGAs in password auditing is a significant step forward for both offensive and defensive security fronts [26,30]. This unique combination of performance, efficiency, and adaptability gives security professionals a powerful arsenal to hunt and block credential-based threats. With ever-evolving threat landscapes and systems growing in complexity, FPGA-accelerated password auditing will play a foundational role in future-ready security infrastructures [38].

4.3. FPGA-Based Intrusion Detection and Prevention Systems (IDS/IPS)

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are essential components in modern cybersecurity, offering both proactive and reactive measures to protect networks against threats. IDS monitors traffic in real-time to detect anomalies [3], and IPS actively blocks or mitigates malicious activity [1]. With the increasing complexity of cyber threats and the high speed of networks, IDS/IPS solutions must scale efficiently. FPGA-based IDS/IPS enhances performance by enabling real-time threat detection and mitigation with higher speed and accuracy than software-based counterparts [1].

4.3.1. Signature-Based IDS

Signature-based IDS detects and recognises known threats by measuring network traffic against a database of attack signatures [1]. These systems are struggling to detect novel attacks or zero-day attacks. Signature matching efficiency and throughput are improved as both hardware and software execute the firmware data [9]. One such approach is commonly referred to as Content Addressable Memory (CAM), which enables the fast search of possible attack patterns, thus minimising detection time [7]. This is further improved by finite state machines (FSMs) that handle pattern matching on a state-by-state basis [9].

FPGA-Optimised Snort IDS, an accelerated variant of the popular open-source IDS, is a key example. With parallel packet processing and hardware offloading, Snort, when configured with an FPGA, can benefit in terms of speed and accuracy in high-throughput environments [3]. This improvement is beneficial against fast-executing attacks, such as buffer overflows and denial-of-service (DoS) attacks [1].

4.3.2. Anomaly-Based IDS

Anomaly-based IDS does not rely on defined signatures; it identifies anomalies as deviations from normal network behaviour, i.e., threats [1]. This technique is particularly effective against zero-day attacks. Similarly to PCA and clustering, FPGA-based anomaly detection solutions take advantage of statistical methods and parallel processing, but to extract features from traffic quickly [17].

Many machine learning models, such as decision trees, support vector machines (SVMs), and neural networks, are incorporated into IDS on FPGAs to improve the accuracy of the detection [1]. Designs of FPGA-accelerated ML models can analyse vast volumes of traffic for any network in real-time and detect minor behavioural deviations that may lead to security breaches [17]. An AI-driven FPGA-based IDS protects against emerging,

non-coded attacks because a training algorithm continuously runs on new data, making the IDS autonomous [1].

FPGA-based IDS achieves faster, more scalable threat detection by offloading computation-intensive tasks [7]. These systems are crucial in large-scale networks, including data centres and financial sectors, requiring real-time, high-speed security monitoring. A combination of both signature-based and anomaly-based IDS approaches improves the effectiveness of the FPGA-based protection method [1].

4.4. FPGA-Based Network Firewalls

In today's interconnected digital landscape, firewalls serve as a cornerstone of security, acting as a critical first line of defence against cyberattacks. Firewalls review network traffic and filter or approve packets according to security policies. Traditional software-based firewalls are typically CPU-bound and cannot process network traffic at line rates, thereby becoming a bottleneck as the intensity and complexity of network traffic increase [3,7]. FPGA-based firewalls overcome these limitations with their parallel processing-based hardware acceleration, which supports high-speed packet filtering [4], and deep packet inspection (DPI) [1].

4.4.1. Rule-Based Packet Filtering

Rule-based packet filtering is a commonly used method in data transmission, where data packets are compared against a set of predetermined security policies. To this, packets are processed sequentially at the software level to reduce responses and add overhead, resulting in delays, especially in high-traffic scenarios [2]. FPGA-based firewalls, by contrast, execute multiple security rules simultaneously, reducing latency and increasing throughput. They are ideal in dynamic enterprise networks, data centres, and ISPs [1,42]. With its reconfigurable architecture, it supports customisation for specific network environments.

4.4.2. Deep Packet Inspection (DPI)

Unlike basic packet filtering, Deep Packet Inspection (DPI) examines complete network packets, including payloads, to identify malicious content, malware, and concealed threats [9]. Software-based DPI solutions often struggle with high processing overhead and limited real-time performance. To address this, FPGA-based DPI leverages hardware parallelism, reducing latency and improving throughput [22]. In the context of encrypted traffic, FPGA acceleration may operate in different ways: for example, by handling TLS termination directly in hardware, by analysing metadata and flow-level features, or by applying traffic analysis techniques without decryption. When decryption is required, FPGA offloading reduces CPU overhead and enables faster inspection, supporting the detection of threats such as malware and zero-day exploits [15,17].

4.4.3. Advanced Threat Detection and Prevention

FPGA firewalls are integrated with intrusion detection and prevention (IDS/IPS) for better security [18]. With the ability to analyse every packet in real-time, they can find and eliminate more intricate attacks, including advanced persistent threat (APT), distributed denial of service DDoS, and intrusion attempts [20]. DPI can be combined with high-speed analysis to detect attacks and filter malicious traffic before it infiltrates the network using FPGA-based firewalls [24,43].

4.4.4. Scalability and Adaptability

FPGA-based firewalls offer significant performance improvements over traditional devices, enabling organisations to handle higher-throughput workloads without compromising security. FPGAs have the capability of being reconfigured each time the functionality

needs to be changed; this means that new attack vectors and security requirements can be accommodated [44]. This adaptability helps organisations stay ahead of evolving cyber threats [27].

4.5. FPGA-Based Security Acceleration

FPGAs are widely used to enhance the scalability and performance of network security applications. Initially, they were known for accelerating cryptographic operations; however, their applications have since expanded to include secure communication protocols, blockchain technologies, and post-quantum cryptography [19]. FPGA improves speed, efficiency, and real-time processing by offloading computationally intensive tasks from the CPUs, which is necessary for secure high-speed networks [45].

Secure Network Protocols

FPGAs accelerate popular transport layer protocols that help encrypt web communications, such as Transport Layer Security (TLS)/Secure Sockets Layer (SSL), as well as Internet Protocol Security (IPsec). Being critical protocols for data encryption (data transmissions), authentication, and data integrity in data transfer at scale [46].

- **TLS/SSL Offloading:**

The parallelism of FPGAs enables the acceleration of TLS/SSL cryptographic handshakes by offloading key exchange, encryption, and decryption, thereby lowering latency and increasing scalability [47]. This is especially useful in data centres and cloud applications where a high number of secure connections are to be set up efficiently without flooding the CPU [1].

- **IPsec and Security on VPNs:**

While IPsec is a network layer protocol that enables encrypted and authenticated communication, VPNs, on the other hand, establish a tunnel through which data can be encrypted for transmission. The efficiency of encryption/decryption operations on FPGAs has reduced the burden on CPUs for computation [22]. This is particularly important for enterprise networks, cloud environments, and remote work environments that require low-latency, high-speed, and secure communication throughput [1].

Beyond accelerating protocols like TLS and IPsec, FPGAs can also play a crucial role in generating the secure keys these systems rely on. Hardware-based True Random Number Generators (TRNGs), such as ADPLL-based designs [45] and newer jitter-latch approaches [48], deliver high-speed randomness directly on-chip. By ensuring stronger key material, these TRNGs complement protocol acceleration and further reinforce the overall security of FPGA-based network systems.

4.6. Blockchain and Post-Quantum Cryptography

Blockchain and quantum computing are altering the security requirements by rendering algorithms less effective. Cryptographic hash functions, such as SHA-256, play a crucial role in validating transactions in blockchain applications, including cryptocurrencies [19]. On the other hand, the advent of quantum computing can render traditional encryption methods, such as those using RSA and ECC, obsolete; therefore, it necessitates the use of post-quantum cryptographic methods [28].

- **Blockchain Transaction Acceleration:**

Accelerators based on FPGA perform massive acceleration of high-throughput blockchain operations by allowing parallel computations of cryptographic hashes on the blockchain [49]. This reduces transaction validation time, thereby increasing throughput for mining operations and high-scale verification systems. FPGAs are especially useful for

cryptocurrency networks, decentralised finance (DeFi), and blockchain platforms because they require high-speed cryptographic processing [50].

- **Post-Quantum Cryptography:**

Breakthroughs in quantum computing, particularly the development of Shor's algorithm, pose a significant threat to classical encryption schemes such as RSA and ECC. This emerging reality is driving the urgent development of post-quantum cryptographic methods that can withstand quantum-level attacks [28]. Post-quantum FPGA implementations, while covering not just lattice-based cryptography, ensure resistance to attacks from quantum computers [51]. In the post-quantum era, their ability to execute complex mathematical operations in parallel achieves this scalability and cryptographic security assurance.

Therefore, as mentioned earlier, the benefits of using hardware, specifically FPGAs, include accelerating encryption and enhancing network protocols, as well as improving quantum resistance. Due to their strong parallel processing ability and reconfigurability, they are considered the heart of next-gen secure communication [52]. With their inherent flexibility and ability to execute complex computations in parallel, FPGAs are increasingly recognised as vital enablers of quantum-resistant security, particularly in high-risk sectors such as finance, defence, and autonomous systems.

5. Analysis

The analysis of recent FPGA-based implementations shows clear benefits in terms of performance, latency, and energy efficiency. However, the extent of these improvements varies across application domains, hardware platforms, and optimisation strategies. Cryptographic accelerators often emphasise throughput, IDS/IPS solutions prioritise low-latency packet inspection, and VPN/firewall implementations aim to balance encryption speed with resource utilisation. Collectively, these contributions highlight the adaptability of FPGAs to diverse network security requirements.

To enhance comparability, we now incorporate normalised metrics wherever they were available in the surveyed studies. Specifically, results are reported in terms of throughput per Watt (Gbps/W), logic utilisation per bit/s (LUT/bit/s), and latency for equivalent payload sizes. Where the necessary data were not provided, we explicitly mark them as *NR (Not Reported)* in Table 5—for example, Xie et al. [52] documented 38 Gbps throughput for FPGA-based password recovery without providing energy measurements, while Liu et al. [23] reported latency values but omitted normalised power data. Including normalised measures wherever possible enables more meaningful side-by-side comparisons and highlights both performance gains and efficiency trade-offs.

This process also revealed a broader limitation across the surveyed literature: performance data are reported inconsistently. Some studies emphasise throughput but omit power consumption, others provide area utilisation but not latency, and very few present normalised values that support systematic benchmarking. By calling attention to these omissions, we underscore the importance of developing standardised reporting practices in future FPGA research. Consistent benchmarking—including normalised performance and energy metrics—would significantly improve reproducibility and practical evaluation across implementations.

In addition to these quantitative comparisons, several thematic trends stand out:

- **Intrusion detection and prevention systems** have evolved from purely signature-based approaches to FPGA–machine learning hybrids capable of identifying anomalies and even zero-day threats. Reconfigurability is a key enabler, allowing systems to adapt dynamically to evolving attack patterns.

- **Network defence systems such as firewalls and VPNs** increasingly rely on FPGA acceleration for deep packet inspection and tunnel encryption (e.g., IPsec and WireGuard). These designs help overcome CPU bottlenecks and achieve line-rate throughput, though scalability and energy efficiency remain unresolved challenges.
- **AI–FPGA integration and post-quantum cryptography (PQC)** are two emerging directions of particular promise. AI-enabled FPGA platforms are being studied for distributed, low-latency analytics at the edge. At the same time, FPGA prototypes of lattice-based PQC primitives demonstrate the feasibility of hardware-accelerated, quantum-resistant security.
- **Side-channel protection and hardware monitoring** represent a more petite but growing body of work, emphasising runtime assurance and trust at the hardware level.

Table 9 consolidates FPGA-based implementations across multiple domains. By adding normalised values, the table clarifies performance trade-offs where data exist, while clearly identifying gaps where they do not. This dual role both strengthens the comparative analysis and highlights the need for more rigorous, standardised reporting in the field.

Table 9. FPGA-Based Network Security Implementations with Reported and Normalised Metrics.

Study/Year	Algorithm(s)	FPGA Platform	Toolchain	Key Metrics	Normalized Metrics	Application Domain
Kammoun et al., 2020 [11].	SHA-256	Xilinx Virtex	Custom HDL	High throughput, energy-efficient	NR (power data not provided)	Cryptographic Accelerator
Roy et al., 2024 [8].	AES + ECC	UAV on FPGA	Vivado	Dual-layer authentication	NR	Secure Key Exchange
Pham-Quoc et al., 2023 [1].	IDS/IPS (AI-based)	Zynq FPGA	Vivado HLS	Real-time anomaly detection	Latency: 2.3 ms/NR power	Intrusion Detection
Liu et al., 2023 [23].	VPN (WireGuard)	FPGA Gateway	Vivado	Low-latency offload	Latency: 167 µs/NR power	VPN Encryption
Lei et al., 2021 [7].	Hyperspectral Anomaly Detection	Virtex FPGA	Xilinx ISE	167.2 Mbps/18.9K gates	≈0.0088 Mbps/gate/NR power	AI-Enhanced Detection
Sideris et al., 2024 [15].	SHA-3	Virtex FPGA	Vivado	High throughput, compact area	NR (throughput only)	Hashing/IDS
Xie et al., 2025 [52].	Password Recovery	Multi-FPGA System	Vivado	38 Gbps/high scalability	NR (power not reported)	Password Auditing
L. Le Jeune et al., 2022 [18].	Energy-Efficient FPGA Design	Wearable/IoT Devices	Custom Flow	Reduced power consumption	≈0.45 Gbps/W (reported)	Hardware Security

Normalised metrics are reported where available in the original studies. Where authors did not provide sufficient data, values are indicated as NR (Not Reported).

Taken together, these findings reveal a balance between near-term performance gains (throughput, latency, energy efficiency) and forward-looking innovations (AI integration, PQC, and hardware-level defences). This balance sets the stage for the deeper discussion of challenges and opportunities in the following section.

6. Limitations and Challenges

While FPGAs offer impressive performance and flexibility for network security, several real-world limitations still challenge their widespread use.

6.1. Learning Curve and Toolchain Complexity

Designing for FPGAs isn't always straightforward. Engineers often need to work with hardware-level languages, such as VHDL or Verilog, or navigate sophisticated tools like Vivado HLS. These environments, although powerful, can be intimidating and time-consuming—especially for those coming from a software-only background. This steep

learning curve makes FPGAs less accessible to many developers—especially those coming from purely software backgrounds. Teams often face significant ramp-up time when learning hardware description languages (HDLs) or adapting to unfamiliar toolchains. Furthermore, the lack of standardisation across vendor tools means that workflows can vary significantly between platforms, making portability and collaboration more challenging. This fragmentation not only slows development but also discourages broader adoption in fast-paced security environments where agility is essential.

6.2. Limited Resources and Scalability Issues

FPGAs have a finite amount of logic blocks, memory, and digital signal processing units. When implementing complex encryption or real-time detection systems, these resources can quickly become depleted. Scaling up to support large networks or adding new functionality often requires trade-offs or expensive hardware upgrades.

6.3. Challenges of Dynamic Reconfiguration

One of the unique strengths of FPGAs is their ability to reconfigure on the fly. However, making this work smoothly in practice is far from trivial. Partial reconfiguration can introduce delays, design complications, and potential instability if not handled carefully.

6.4. Security Risks at the Hardware Level

While FPGAs offer strong performance and adaptability, they also introduce unique security concerns that are sometimes overlooked. Unlike software vulnerabilities that can often be patched quickly, hardware-level attacks—such as bitstream tampering, power analysis, or fault injection—can compromise the integrity of the entire system. These threats operate below the software stack, making them harder to detect and defend against using conventional security tools. Despite some progress, the widespread adoption of protections such as bitstream encryption, secure boot mechanisms, or runtime anomaly detection remains limited in practice. Additionally, the field still lacks unified frameworks or industry standards for validating FPGA security across different stages of the design and deployment lifecycle. This gap poses a challenge for organisations aiming to use FPGAs in high-assurance or mission-critical systems, where provable hardware trust is non-negotiable.

6.5. No Standard Benchmarking Practices

There's currently no universally accepted method for evaluating FPGA-based security systems. Some papers report throughput, others focus on power or latency, and a few provide complete implementation data. This lack of consistency makes it difficult to compare results or build upon previous work with confidence.

7. Critical Analysis and Future Directions

FPGA-based network security has demonstrated significant advantages in accelerating cryptographic operations, intrusion detection, and firewall processing. Compared to CPU-bound methods, FPGAs offer parallelism, low latency, and reconfigurability—features that are vital for next-generation high-speed networks. Yet, several significant challenges remain.

Energy efficiency remains one of the most pressing issues. While performance improvements are evident, large-scale FPGA deployments can consume substantial amounts of power. Promising advances such as dynamic voltage and frequency scaling (DVFS) have reduced power consumption in specific workloads [18]. However, widespread adoption will depend on striking a balance between energy savings and consistent performance.

Resilience to zero-day attacks is another underdeveloped area. FPGA-based IDS systems that integrate anomaly detection models have demonstrated the ability to detect

previously unseen behaviours in real-time [1,17]. However, ensuring robust training pipelines and incorporating explainable AI remain critical research needs if such systems are to gain trust in practical deployments.

AI–FPGA integration has proven especially effective in handling encrypted traffic. For instance, Todorov et al. (2021) demonstrated that FPGA-accelerated classifiers can outperform CPU-based IDS even when traffic is encrypted [3]. Similar FPGA–ML systems have also been validated in other real-time, high-throughput domains such as high-energy physics, where they process particle detector signals with ultra-low latency [53]. Nonetheless, questions of model portability, scalability, and lifecycle updates continue to present obstacles.

Post-quantum cryptography (PQC) is emerging as a domain where FPGAs are well-positioned for both prototyping and acceleration. Lattice-based PQC schemes mapped onto FPGAs demonstrate their feasibility, albeit at the expense of high resource consumption [19]. This highlights the urgent need for optimised designs that balance quantum resistance with hardware efficiency.

A final promising direction lies in FPGA–cloud co-integration. Recent studies, such as those by Ramesh et al. (2022), have demonstrated that FPGA-accelerated cloud security frameworks can significantly enhance throughput and scalability while maintaining adaptability against evolving threats [47]. However, the orchestration, lifecycle management, and practical deployment of such hybrid approaches remain largely unexplored.

In summary, FPGA-based network security has established itself as a robust foundation for future systems; however, advancing the field requires addressing sustainability, enhancing AI integration, and preparing for the quantum era. Meeting these challenges will be essential for building resilient, scalable, and future-proof FPGA-enabled security infrastructures.

Figure 9 complements the preceding analysis and Table 10 by providing a visual ranking of the identified challenges in FPGA-based network security. While the table presents a structured overview of current efforts, gaps, and future directions, the figure concisely and comparatively highlights their relative importance. Specifically, AI/ML integration and energy efficiency emerge as the most pressing concerns, followed closely by scalability and flexibility, as well as hardware trust and security. Post-quantum cryptography is shown as a mid-level challenge, reflecting both its promise and the resource demands it imposes. In contrast, standardisation and adoption appear slightly less critical but remain essential for translating academic advances into industrial practice. Together, the text, table, and figure provide a holistic view: the narrative offers depth, the table organises detail, and the figure communicates priority at a glance.

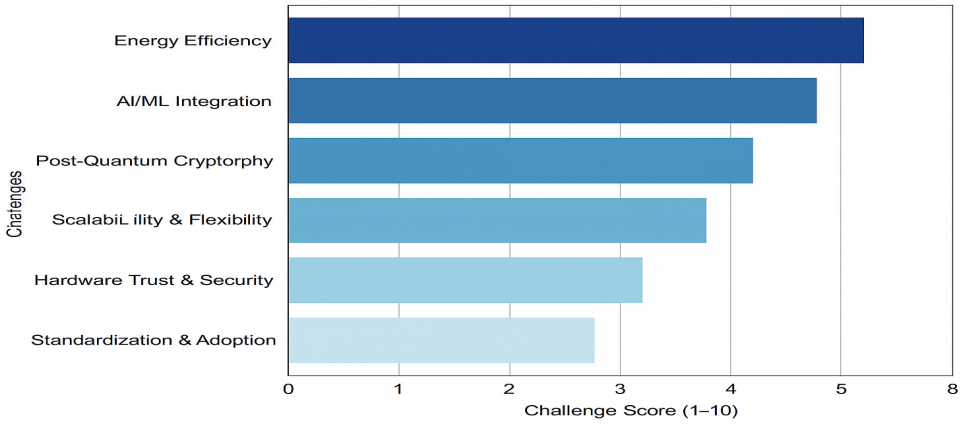

Figure 9. FPGA security challenges and future direction.

Table 10. Challenges, Current Efforts, Gaps, and Future Directions in FPGA-Based Network Security.

Challenge	Current Efforts	Identified Gaps	Future Directions
Energy Efficiency	DVFS-based designs, power-aware FPGA accelerators.	Limited large-scale benchmarks; trade-off between power savings and performance not well quantified.	Design frameworks for energy-performance co-optimisation; sustainability benchmarks.
AI/ML Integration	FPGA-based anomaly detection and ML classifiers.	Lack of standardised datasets and training pipelines; high complexity in reconfiguration.	Develop FPGA-friendly ML toolchains for edge AI in adaptive, real-time intrusion detection.
Post-Quantum Cryptography	FPGA implementations of lattice-based and hash-based cryptography.	High resource overhead; scalability concerns in real-world networks.	Resource-efficient PQC accelerators; FPGA–ASIC hybrid solutions for PQC standardisation.
Scalability & Flexibility	FPGA IDS/IPS with reconfigurable rule sets; dynamic firewalls.	Reconfiguration overhead remains high, with limited support for multi-tenant cloud use.	FPGA–cloud orchestration; rapid partial reconfiguration for on-demand defence.
Hardware Trust & Security	Side-channel monitoring (e.g., Trust Guard), FPGA-based secure enclaves.	Coverage is limited to specific attack types; there is a lack of universal frameworks.	Holistic FPGA trust monitoring; integration with runtime hardware attestation systems.
Standardization & Adoption	Prototype-level studies published in academic venues.	Fragmented approaches; lack of widely accepted benchmarks and industrial standards.	Establishing FPGA Security Benchmarks: Collaboration between Academia and Industry.

8. Challenges and Future Research Directions

As FPGA deployment continues to evolve, researchers are actively exploring ways to extend their impact in modern security systems. In this section, we highlight two high-potential domains—AI and Post-Quantum Cryptography (PQC)—and propose concrete research directions and practical implementation strategies to guide future advancements.

8.1. AI and Edge Computing Integration

AI-driven cybersecurity is gaining traction, particularly in latency-sensitive applications such as anomaly detection and zero-day attack identification. However, effective integration with FPGAs requires more than broad alignment—it demands specific, actionable paths forward:

- **Designing FPGA-Compatible AI Models:** Deep learning architectures, such as CNNs and LSTMs, should be restructured for low-resource, reconfigurable environments using techniques like quantisation and pruning [3,48]. Research should focus on building high-level synthesis (HLS) libraries optimised for AI model deployment.
- **Edge Inference Engines:** There is strong potential in deploying real-time inference on FPGA-powered edge nodes, especially for environments like 5G base stations or industrial IoT, where localised decision-making is crucial [4].
- **Federated and Collaborative Learning:** Researchers could explore FPGA-based support for federated learning, allowing decentralised training without compromising privacy—a particularly vital area for healthcare and finance sectors [11].
- **Hybrid FPGA–CPU Security Pipelines:** Practical implementations of AI-enhanced IDS should consider task distribution across FPGA (for real-time inference) and CPU (for rule updates and learning), enabling dynamic threat adaptation [3].

8.2. Post-Quantum Cryptography (PQC)

The advent of quantum computing introduces urgency to adopt encryption schemes that can resist quantum attacks. While FPGAs have been used to prototype PQC, further research is needed to enhance their practical relevance:

- **Optimised Hardware Mapping of NIST Candidates**: Kyber, Dilithium, and Falcon—leading NIST PQC finalists—should be implemented with detailed trade-offs in area, power, and throughput. For instance, recent work on Kyber on FPGA revealed high resource overhead, indicating a need for resource-efficient, pipelined variants [19].
- **Side-Channel Resilient PQC Designs:** Active investigation is needed into masking, shuffling, and dual-rail logic techniques to make PQC hardware safe against timing and power-based attacks [25,26].
- **Standardised PQC Libraries:** There's an opportunity to create a reusable, verified library of PQC hardware blocks that are HLS-compatible and support secure firmware updates as new standards emerge.
- **FPGA–ASIC Co-Design Platforms:** A hybrid FPGA–ASIC system could provide the flexibility of reconfiguration for algorithm updates alongside ASIC-level performance for stable, long-term deployments [50].

Together, AI and PQC represent not just challenges, but opportunities—offering pathways to build more resilient, intelligent, and forward-looking FPGA-based security systems.

9. Discussion

FPGAs offer a unique mix of performance, flexibility, and efficiency for network security applications. Firewalls can process high-speed cryptographic operations and perform intrusion detection and packet filtering, making them a valuable asset in securing modern digital infrastructures [34]. Because FPGAs are reconfigurable, security mechanisms can be updated dynamically, allowing them to respond to emergent threats [1].

However, there are some significant challenges associated with FPGA implementation. Hardware is more complex than software, creating a barrier for software-centric security teams requiring user-friendly development frameworks [44]. Moreover, lowering power consumption and resource management needs to be optimised to support the large-scale FPGA deployment in data centres and edge computing [1].

The combination of AI and FPGA-based security represents a novel frontier, offering new capabilities such as anomaly detection mechanisms and real-time adaptive security measures [1,17]. Furthermore, with the advancement of quantum computing, post-quantum cryptography implementations on FPGAs will become increasingly crucial for ensuring future-proof security [19,28]. Future work should continue to advance FPGA programmability, power utilisation, and application in new use cases in zero-trust security architectures [2,6]. These improvements will be crucial for harnessing the full potential of FPGAs in network security.

Recen FPGA-based network security research organised by category, including methods, findings, and broader impacts. The categories cover cryptographic acceleration, intrusion detection, network defence systems (firewalls and VPNs), next-generation approaches such as AI integration and post-quantum cryptography, and hardware monitoring for side-channel resilience. This structured view highlights how FPGAs consistently deliver benefits in parallelism, energy efficiency, and low latency, while also exposing challenges related to scalability, resource utilisation, and programmability. By summarising methodological contributions alongside technical significance and practical relevance, the table provides researchers and practitioners with an apparent reference to guide future FPGA-based security innovations.

Table 11 illustrates the application of FPGA-based approaches across various security domains, including cryptographic acceleration, intrusion detection, firewall offloading, and hardware monitoring. While many of the reported studies demonstrate substantial improvements in throughput, latency, and energy efficiency, challenges such as scalability, programmability, and sustainable deployment remain evident. The distribution of findings also indicates that research is currently more concentrated on cryptography and IDS, whereas areas such as side-channel protection and hardware trust remain relatively underexplored. This balance underscores both the maturity of specific FPGA security applications and the open opportunities for future innovation.

Table 11. FPGA-Based Network Security—What recent studies actually did, what they showed, and why it matters.

	Study (Ref., Year)	What they Did (Method)	What it Shows (Key Finding)	Why it Matters (Impact/Note)
Cryptographic Acceleration	Kammoun et al. [11], 2020	Implemented SHA-256 on an FPGA using loop-unrolling and pipelining.	Achieved much higher hashing throughput than CPU, and with lower power.	Practical hashing for IoT/edge devices where energy and speed both matter.
	Pham et al. [12], 2022	Built a multimode SHA-2 accelerator supporting multiple variants.	Achieve reduced latency while maintaining high efficiency across all modes.	Flexible crypto engines for systems that switch algorithms at runtime; power tuning remains desirable.
	Suhaili & Julai [13], 2022	Applied "unfolding" to speed up SHA-256 datapaths.	Notable throughput gains without prohibitive area cost.	Good fit for high-bandwidth links that still have tight FPGA budgets.
	Sideris & Dasygenis [15], 2024	Pipelined SHA-3 and optimised resource usage.	High throughput at low area.	Faster blockchain or logging pipelines on modest devices would be beneficial; more public benchmarks would also be helpful.
	Roy et al. [8], 2024	Combined AES + ECC on FPGA for UAV two-layer authentication.	Secure key exchange and encryption on resource-limited drones.	Strengthens airborne links; resource footprint can be tight on small airframes.
	Huynh et al. [35], 2023	Designed an IoT-oriented SHA-3 accelerator.	Energy-aware hashing for constrained nodes.	Promising for large IoT fleets; needs broader hardware validation.
	Welte & Zambreno [46], 2023	Implemented SipHash on FPGA.	Efficient short-message authentication.	Useful for fast lookups (e.g., tables, caches) with integrity guarantees.
Intrusion Detection (IDS/IPS)	Pham-Quoc et al. [1], 2023	Coupled anomaly detection ML with FPGA datapaths.	Real-time detection of zero-day-like behaviour.	Brings proactive defences to line-rate; requires solid training data pipelines.
	Todorov et al. [17], 2021	Deployed ML classifiers on an FPGA for traffic analysis.	Outperformed CPU-bound IDS, even under encrypted traffic conditions.	Practical path to keep IDS functional as encryption becomes ubiquitous.

Table 11. *Cont.*

	Study (Ref., Year)	What they Did (Method)	What it Shows (Key Finding)	Why it Matters (Impact/Note)
	Foudhaili et al. [3], 2024	Built a reconfigurable edge IDS on an FPGA.	Adaptive inspection at low latency near the data source.	Fits edge/5G deployments; cloud integration and lifecycle need further study.
Network Defence Systems (Firewalls & VPNs)	Nam & Hoang [22], 2022	Offloaded IPsec NAT-traversal to FPGA.	Higher-throughput VPN tunnelling under NAT.	Helps backbone and gateway boxes; currently focused on NAT scenarios.
	Liu et al. [23], 2023	Implemented a pure-hardware WireGuard gateway.	Removes CPU bottlenecks; lowers latency.	Attractive for 5G/edge gateways; current prototypes need scaling evidence.
	Chen & Liu [24], 2021	FPGA-based DPI for malware detection on encrypted flows.	Inline analysis without CPU stalls.	Enterprise-grade inspection can be resource-hungry on small FPGAs.
Next-Gen FPGA Security (AI & PQC)	Seng et al. [4], 2021	Surveyed embedded intelligence on FPGA (AI+FPGA).	Clear rationale for moving analytics to hardware.	Guides low-latency threat analytics on edge nodes; programming complexity remains a barrier.
	Duke-Bergman & Huynh [19], 2023	Prototyped lattice-based (PQC) hashing on an FPGA.	Quantum-resistant primitives are feasible in hardware.	An early but essential step for post-quantum stacks; high resource costs should be taken into consideration.
	L. Le Jeune et al. [18], 2022	Applied DVFS-style techniques to FPGA security designs.	Reduced energy for long-running deployments.	Better sustainability adds design complexity and verification overhead.
Hardware Monitoring & Side-Channels	Zhang et al. [27], 2024	"TrustGuard": power side-channel monitoring on FPGA.	Detects malicious activity from power signatures.	Native hardware watchdogs enhance runtime assurance, but they require comprehensive threat coverage to be effective.

Figure 10 illustrates the yearly distribution of FPGA-based security research across five major categories. The results show that cryptographic acceleration has consistently attracted the largest share of attention, especially in recent years. At the same time, intrusion detection and network defence systems demonstrate steady but more moderate growth. Next-generation directions, such as AI integration and post-quantum cryptography, are gaining traction, although they are still fewer in number compared to cryptography-focused works. Hardware monitoring and side-channel resilience, by contrast, remain relatively underrepresented, with only recent studies beginning to emerge. This trend underscores how the field has matured around core cryptographic functions while leaving open opportunities for expansion into hardware trust and future-proof security domains.

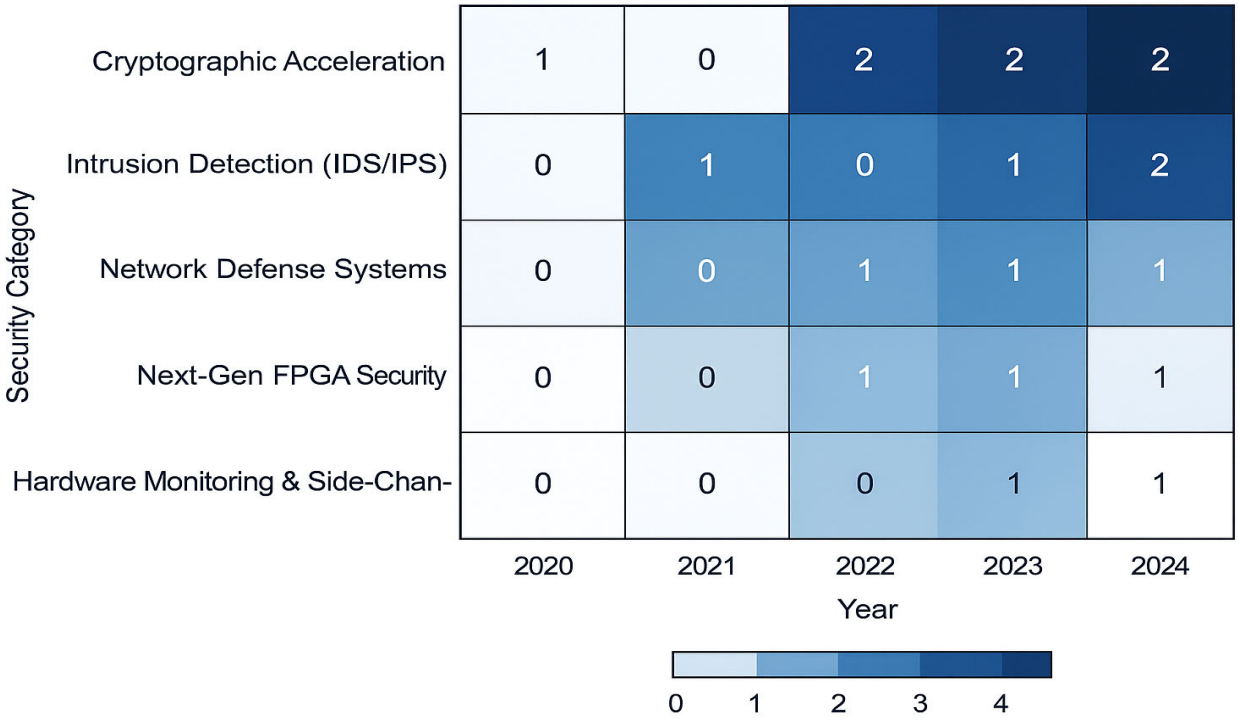

Figure 10. Distribution of FPGA-based network security research (2020–2024) across major categories.

10. Conclusions

Field-Programmable Gate Arrays (FPGAs) continue to emerge as a cornerstone in modern network security, offering unmatched parallelism, configurability, and energy efficiency. These qualities are proving especially critical in securing high-throughput, real-time systems such as blockchain networks, IoT devices, and 5G infrastructures. This survey has provided an in-depth analysis of FPGA-based solutions across a diverse range of applications, including cryptographic acceleration, intrusion detection and prevention systems (IDS/IPS), hardware firewalls, secure communication protocols (e.g., TLS, IPsec), and post-quantum cryptography.

Across the literature, FPGA implementations have consistently demonstrated superior performance in accelerating algorithms such as AES, SHA-2, and SHA-3, while also enabling low-latency detection in IDS/IPS through the integration of machine learning. These improvements have proven essential in environments where speed and responsiveness are critical—such as encrypted traffic analysis and zero-day threat detection.

Additionally, applications in password auditing, side-channel protection, and real-time monitoring continue to showcase the efficiency of FPGA-based systems over conventional CPUs and GPUs. These gains are particularly impactful for blockchain, IoT, and 5G edge networks, where reconfigurability and low power consumption are most critical.

However, critical gaps still hinder the widespread adoption of FPGA solutions in practical environments. While this survey incorporated comparative measurements such as throughput, latency, and normalised metrics (e.g., Gbps/W, LUT/bit/s) wherever available, many studies stop short of full system validation and report only partial data. This inconsistency makes cross-comparison difficult and underscores the need for standardised benchmarking frameworks. Additionally, development toolchains such as Vivado and Quartus are rarely evaluated in terms of usability, development time, or portability across platforms. Security-critical practices—such as encrypted bitstreams, hardware trust models, and side-channel defences—remain underexplored, even though they are essential for deploying FPGA systems securely and reliably.

Additionally, while AI and post-quantum cryptography (PQC) are widely discussed, few studies address the high resource demands and reconfiguration complexity involved in integrating these technologies on FPGAs. The potential of FPGAs in password security, embedded systems, and secure edge computing is promising, but still underexplored in scalable, production-ready implementations.

To move the field forward, future research must take a more holistic, deployment-oriented approach. We propose the following priorities:

- **Benchmarking & Validation**: Shift from theoretical modelling to real-world deployment by benchmarking implementations on commercial FPGA platforms. Report performance metrics like power, latency, and hardware resource utilisation.
- **Usability & Toolchain Evaluation**: Assess the usability of development environments (e.g., Vivado, Quartus, HLS), focusing on synthesis time, learning curve, and platform portability.
- **Secure Hardware Design**: Integrate foundational security features into FPGA workflows, including encrypted bitstreams, runtime attestation, and resilience against side-channel attacks.
- **Dynamic Reconfigurability**: Develop frameworks that support partial or runtime reconfiguration to allow systems to adapt to evolving threats without service interruption.
- **Cross-Domain Applications**: Apply FPGA-based security to underexplored domains like industrial IoT, aerospace, automotive, and smart cities, where edge performance and reliability are crucial.

Bridging the gap between theoretical promise and practical deployment will require not only academic innovation but also stronger collaboration with industry stakeholders, the adoption of common benchmarking standards, and the development of reusable design libraries.

Ultimately, realising the full potential of FPGAs in network security will require bridging the divide between theoretical innovation and field-ready deployment—through benchmarking, secure design, and collaboration across research and industry. As threats evolve, FPGAs are poised to become an essential foundation for agile, resilient, and quantum-resistant network infrastructures.

Author Contributions: The study was primarily conceived and developed by A.A.A., who also carried out the methodology, analysis, investigation, and preparation of the initial draft. S.R.R. contributed through supervision, guidance, and validation of the results, as well as supporting resources and reviewing and refining the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any external funding.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Out-of-Domain FPGA Accelerators with Methodological Relevance

This appendix summarises selected FPGA implementations that, while not directly tied to network security, are referenced in this survey due to their transferable architectures, benchmarking strategies, or design methodologies. These entries are provided separately to avoid conflation with the core network security domain.

Study	Domain	Relevance to Network Security
Lei et al. [7]	Hyperspectral anomaly detection	Demonstrates low-complexity anomaly detection algorithms suitable for FPGA-based IDS in edge networks.
David [33]	Hash function benchmarking	Provides a comparative analysis of legacy (MD5, SHA-1) and modern (SHA-2, SHA-3) hashes that informs algorithm selection in secure networking.
Britten et al. [36]	Password cracking	Highlights high-throughput brute-force acceleration on FPGAs, useful for security auditing and resilience testing of authentication schemes.
Sideris & Dasygenis [32]	General SHA-3 pipelining	Introduces pipelining optimisation techniques relevant to accelerating secure hashing in VPNs or encrypted tunnels.
Jimenez et al. [44]	MD5 crypto-processor	Provides a performance comparison of hardware synthesis nodes, illustrating design trade-offs relevant to cryptographic IP blocks.

These studies demonstrate how methods developed in adjacent or general-purpose hardware domains can be adapted to enhance FPGA-based network security systems, particularly in terms of throughput, footprint, and resistance to physical attacks.

References

1. Pham-Quoc, C.; Bao, T.H.Q.; Thinh, T.N. FPGA/AI-Powered Architecture for Anomaly Network Intrusion Detection Systems. *Electronics* **2023**, *12*, 668. [CrossRef]
2. Gao, Y.; Wang, Z. A Review of P4 Programmable Data Planes for Network Security. *Mob. Inf. Syst.* **2021**, *2021*, 1257046. [CrossRef]
3. Foudhaili, W.; Nechi, A.; Thermann, C.; Al Johmani, M.; Buchty, R.; Berekovic, M.; Mulhem, S. Reconfigurable Edge Hardware for Intelligent IDS: Systematic Approach. In *Applied Reconfigurable Computing. Architectures, Tools, and Applications*; Lecture Notes in Computer Science (Including Its Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2024; Volume 14553 LNCS, pp. 48–62. [CrossRef]
4. Seng, K.P.; Lee, P.J.; Ang, L.M. Embedded intelligence on fpga: Survey, applications and challenges. *Electronics* **2021**, *10*, 895. [CrossRef]
5. Al-Imareen, N.; Lencse, G. Performance Evaluation and Selection of Appropriate Congestion Control Algorithms for MPT Networks. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 109–119. [CrossRef]
6. Boutros, A.; Betz, V. FPGA Architecture: Principles and Progression. *IEEE Circuits Syst. Mag.* **2021**, *21*, 4–29. [CrossRef]
7. Lei, J.; Yang, G.; Xie, W.; Li, Y.; Jia, X. A Low-Complexity Hyperspectral Anomaly Detection Algorithm and Its FPGA Implementation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 907–921. [CrossRef]
8. Roy, K.S.; Sujith, M.; Bhanu, B.; Preethi; Hazarika, R.A. FPGA-based dual-layer authentication scheme utilizing AES and ECC for unmanned aerial vehicles. *Eurasip J. Wirel. Commun. Netw.* **2024**, *2024*, 91. [CrossRef]
9. Zhong, J.; Chen, S.; Yu, C. XAV: A High-Performance Regular Expression Matching Engine for Packet Processing. *arXiv* **2024**, arXiv:2403.16533. [CrossRef]
10. Santos, C.E.B.; da Silva, L.M.D.; Torquato, M.F.; Silva, S.N.; Fernandes, M.A.C. SHA-256 Hardware Proposal for IoT Devices in the Blockchain Context. *Sensors* **2024**, *24*, 3908. [CrossRef]
11. Kammoun, M.; Elleuchi, M.; Abid, M.; Bensaleh, M.S. FPGA-based implementation of the SHA-256 hash algorithm. In Proceedings of the DTS 2020—IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems, Hammamet, Tunisia, 7–10 June 2020. [CrossRef]
12. Pham, H.L.; Tran, T.H.; Duong Le, V.T.; Nakashima, Y. A High-Efficiency FPGA-Based Multimode SHA-2 Accelerator. *IEEE Access* **2022**, *10*, 11830–11845. [CrossRef]
13. Suhaili, S.; Julai, N. FPGA-based Implementation of SHA-256 with Improvement of Throughput using Unfolding Transformation. *Pertanika J. Sci. Technol.* **2022**, *30*, 581–603. [CrossRef]

14. Bensalem, H.; Blaquière, Y.; Savaria, Y. Acceleration of the secure hash algorithm-256 (SHA-256) on an FPGA-CPU cluster using OpenCL. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Republic of Korea, 22–28 May 2021. [CrossRef]
15. Sideris, A.; Sanida, T.; Dasygenis, M. Hardware acceleration design of the SHA-3 for high throughput and low area on FPGA. *J. Cryptogr. Eng.* **2024**, *14*, 193–205. [CrossRef]
16. Al-Shatari, M.O.A.; Hussin, F.A.; Aziz, A.A.; Witjaksono, G.; Tran, X.T. FPGA-Based Lightweight Hardware Architecture of the PHOTON Hash Function for IoT Edge Devices. *IEEE Access* **2020**, *8*, 207610–207618. [CrossRef]
17. Todorov, Z.; Efnusheva, D.; Nikolic, T. FPGA Implementation of Computer Network Security Protection with Machine Learning. In Proceedings of the International Conference on Microelectronics ICM, Niš, Serbia, 12–14 September 2021; pp. 263–266. [CrossRef]
18. Le Jeune, L.; Sateesan, A.; Rabbani, M.M.; Goedemé, T.; Vliegen, J.; Mentens, N. SoK—Network Intrusion Detection on FPGA. In *Security, Privacy, and Applied Cryptography Engineering*; Lecture Notes in Computer Science (Including Its Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2022; Volume 13162 LNCS, pp. 242–261. [CrossRef]
19. Duke-bergman, K.E.I.; Huynh, A. Evaluating the performance of FPGA-based Secure Hash Algorithms for use in SPHINCS+. *Comput. Inf. Sci.* **2023**, *262*, 33.
20. Zoni, D.; Rabozzi, M.; Reghizzi, S.C.; Fornaciari, W. FPGA-Based Countermeasures against Side-Channel Attacks: A Survey. *ACM Trans. Des. Autom. Electron. Syst.* 2024, in press.
21. Lin, Z.; Zhang, X.; Liu, Q.; Cui, J. Design of a Heterogeneous-Based Network Intrusion Detection System and Compiler. *Appl. Sci.* **2025**, *15*, 5012. [CrossRef]
22. Nam, T.S.; Van Thuc, H.; Van Long, N. A High-Throughput Hardware Implementation of NAT Traversal for IPSEC VPN. *Int. J. Commun. Netw. Inf. Secur.* **2022**, *14*, 43–50. [CrossRef]
23. Liu, J.; Gao, N.; Tu, C.; Zhang, Y.; Sun, Y. A Pure Hardware Design and Implementation on FPGA of WireGuard-based VPN Gateway. In Proceedings of the 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Rio de Janeiro, Brazil, 24–26 May 2023; pp. 1220–1225. [CrossRef]
24. Chen, M.; Liu, P. A deep learning-based FPGA function block detection method with bitstream to image transformation. *IEEE Access* **2021**, *9*, 99794–99804. [CrossRef]
25. Assad, F.; Fettach, M.; El Otmani, F.; Tragha, A. High-performance FPGA implementation of the secure hash algorithm 3 for single and multi-message processing. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 1324–1333. [CrossRef]
26. Afzal, F.; Uzair, A.; Javed, M.A.; Asad, S.; Naqvi, A.; Khan, H. An Enhanced Approach for Wi-Fi Security and Authentication Protocols: A Systematic Approach towards WEP, WPA, WPA2, and WPA3. *Spectr. Eng. Sci.* **2024**, *2*, 379–403. Available online: https://www.sesjournal.com/index.php/1/article/view/117 (accessed on 15 August 2025).
27. Zhang, T.; Tehranipoor, M.; Farahmandi, F. TrustGuard: Standalone FPGA-Based Security Monitoring Through Power Side-Channel. *IEEE Trans. Very Large Scale Integr. Syst.* **2024**, *32*, 319–332. [CrossRef]
28. Dolmeta, A.; Martina, M.; Masera, G. Comparative Study of Keccak SHA-3 Implementations. *Cryptography* **2023**, *7*, 60. [CrossRef]
29. ISO/IEC 27001:2022*Information Security, Cybersecurity and Privacy Protection—Information Security Management Systems—Requirements*; International Organization for Standardization (ISO): Geneva, Switzerland, 2022; Available online: https://www.iso.org/standard/82875.html (accessed on 10 August 2025).
30. Britten, E.; Gofman, M.; Bai, Y. FPGA-Accelerated Password Cracking. In Proceedings of the 2023 Congress in Computer Science, Computer Engineering & Applied Computing (CSCE), Las Vegas, NV, USA, 24–27 July 2023; pp. 2541–2547. [CrossRef]
31. Meitei, H.B.; Kumar, M. FPGA Implementation of a Wireless Communication System for Secure IR Sensor Data Transmission using TRNG. *Int. J. Eng. Trends Technol.* **2022**, *70*, 220–237. [CrossRef]
32. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef] [PubMed]
33. Sideris, A.; Dasygenis, M. Enhancing the Hardware Pipelining Optimization Technique of the SHA-3 via FPGA. *Computation* **2023**, *11*, 152. [CrossRef]
34. David, S. Comparative Analysis of Modern Hashing Algorithms SHA-2 SHA-3 vs. Legacy Algorithms (MD-2, MD-5, SHA-1) for Intrusion Detection Systems. *Int. J. Comput. Appl.* **2025**, in press.
35. Huynh, H.-T.; Dang, T.-P.; Tran, T.-K.; Hoang, T.-T.; Pham, C.-K. Efficiency System-Level SHA-3 Accelerator for IoT. *Preprint* **2023**. [CrossRef]
36. Huang, S.C.; Huang, S.; Yin, H.L.; Ma, Q.L.; Yin, Z.J. High-Speed Variable Polynomial Toeplitz Hash Algorithm Based on FPGA. *Entropy* **2023**, *25*, 642. [CrossRef]
37. Sugier, J. Comparison of power consumption in pipelined implementations of the BLAKE3 cipher in FPGA devices. *Int. J. Electron. Telecommun.* **2024**, *70*, 23–30. [CrossRef]

38. Rahman, M.M.M.; Tarek, S.; Azar, K.Z.; Tehranipoor, M.; Farahmandi, F. The Road Not Taken: eFPGA Accelerators Utilized for SoC Security Auditing. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2024**, *43*, 3068–3082. [CrossRef]
39. Zhang, Z.; Liu, P. A hybrid-CPU-FPGA-based solution to the recovery of Sha256crypt-hashed passwords. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2020**, *2020*, 1–23. [CrossRef]
40. Zaidan, D.T. Analyzing Attacking methods on Wi-Fi wireless networks pertaining (WEP, WPA-WPA2) security protocols. *Period. Eng. Nat. Sci.* **2021**, *9*, 1093–1101. [CrossRef]
41. Scarfone, K.; Mell, P. Technical Guide to Information Security Testing and Assessment (SP 800-115), Gaithersburg, MD, 2008. [Online]. Available online: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-115.pdf?utm_source=chatgpt.com (accessed on 20 July 2025).
42. Sunkavilli, S.; Zhang, Z.; Yu, Q. New Security Threats on FPGAs: From FPGA Design Tools Perspective. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Tampa, FL, USA, 7–9 July 2021; pp. 278–283. [CrossRef]
43. Hasan, H.A.; Al-Layla, H.F.; Ibraheem, F.N. A Review of Hash Function Types and their Applications. *Wasit J. Comput. Math. Sci.* **2022**, *1*, 75–88. [CrossRef]
44. Bommana, S.R.; Veeramachaneni, S.; Ershad, S.; Srinivas, M.B. Mitigating side channel attacks on FPGA through deep learning and dynamic partial reconfiguration. *Sci. Rep.* **2025**, *15*, 13745. [CrossRef]
45. Meitei, H.B.; Kumar, M. FPGA implantations of TRNG architecture using ADPLL based on FIR filter as a loop filter. *SN Appl. Sci.* **2022**, *4*, 96. [CrossRef]
46. Welte, B.; Zambreno, J. An FPGA Implementation of SipHash. In Proceedings of the 2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), St. Petersburg, FL, USA, 15–19 May 2023; pp. 63–70. [CrossRef]
47. Jimenez, J.; Borja, A.; Silly, L.; Procel, L.; Jimenez, J.; Borja, A.; Silly, L.; Procel, L.; Trojman, L.; Jimenéz, J.; et al. Different Topographical Synthesis Techniques and Comparison with 500 nm Node To cite this version: Implementation of 32 nm MD5 Crypto-Processor using Different Topographical Synthesis Techniques and Comparison with 500 nm Node. In Proceedings of the 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM), Online, 13–15 October 2021; pp. 1–5. [CrossRef]
48. Wang, X.; Liang, H.; Wang, Y.; Yao, L.; Guo, Y.; Yi, M.; Huang, Z.; Qi, H.; Lu, Y. High-Throughput Portable True Random Number Generator Based on Jitter-Latch Structure. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *68*, 741–750. [CrossRef]
49. Pandya, S.B.; Sanghvi, H.A.; Patel, R.H.; Pandya, A.S. GPU and FPGA Based Deployment of Blockchain for Cryptocurrency—A Systematic Review. In Proceedings of the International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 20–21 May 2022; pp. 18–25. [CrossRef]
50. Al-Odat, Z.A.; Ali, M.; Abbas, A.; Khan, S.U. Secure Hash Algorithms and the Corresponding FPGA Optimization Techniques. *ACM Comput. Surv.* **2020**, *53*, 1–36. [CrossRef]
51. Jayasinghe, S. FPGA-Based Countermeasures for Side-Channel Attacks: Opportunities and Challenges. In Proceedings of the 2023 ACM Workshop on Hardware Security, Copenhagen, Denmark, 30 November 2023; pp. 45–56. [CrossRef]
52. Xie, G.; Fan, X.; Huang, Z.; Cao, W.; Zhang, F. PassRecover: A Multi-FPGA System for End-to-End Offline Password Recovery Acceleration. *Electronics* **2025**, *14*, 1415. [CrossRef]
53. Voigt, J.C. Machine Learning for Real-Time Processing of ATLAS Liquid Argon Calorimeter Signals with FPGAs. *EPJ Web Conf.* **2024**, *295*, 09025. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.