

Impact of Cipher-Core Selection on XTS-based Secure Network-Attached Storage Systems: A Comparison Study of MKV and AES

DOI: <https://doi.org/10.54654/isj.v1i27.4042>

Tran Van Khanh*, Phan Van Ky, Vu Van Viet

Abstract— This paper presents an FPGA-based, system-level comparative study of MKV and AES block ciphers within a secure Network-Attached Storage (NAS) architecture. While XTS mode is the standard for storage encryption, its hardware impact within real-world architectures is often overlooked. We evaluate MKV, developed by the Vietnam Government Cipher Committee, as a viable alternative to Advanced Encryption Standard (AES) for data-at-rest security. Both algorithms are integrated into an identical XTS-based inline encryption architecture to analyze hardware resource utilization and storage performance under standard benchmark workloads. Results demonstrate that both MKV-XTS and AES-XTS configurations achieve line-rate sequential throughput near SATA Gen 3 limits (~490 MB/s read, ~433 MB/s write at QD=8) with under 2% latency overhead. However, hardware profiling reveals a distinct logic-memory trade-off: the MKV-XTS system requires 19.1% more logic (50,232 LUTs) than the AES baseline, yet it completely eliminates BRAM dependency within the critical data processing path.

Tóm tắt— Bài báo này trình bày nghiên cứu so sánh ở mức hệ thống trên nền tảng FPGA giữa thuật toán mã khối MKV và AES trong kiến trúc lưu trữ mạng (Network-Attached Storage - NAS) bảo mật. Mặc dù XTS là tiêu chuẩn thực tế cho mã hóa lưu trữ, tác động phần cứng của chế độ này trong các hệ thống thực tế thường bị bỏ qua. Chúng tôi đánh giá MKV, do Ban Cơ yếu Chính phủ Việt Nam phát triển, như một giải pháp thay

thế khả thi cho AES đối với bảo mật dữ liệu lưu trữ. Hai thuật toán được tích hợp vào cùng một kiến trúc mã hóa inline dựa trên XTS để phân tích việc sử dụng tài nguyên phần cứng và hiệu năng lưu trữ thông qua các workload tiêu chuẩn. Kết quả cho thấy cả hai cấu hình MKV-XTS và AES-XTS đều đạt thông lượng tuần tự gần giới hạn SATA Gen 3 (~490 MB/s đọc, ~433 MB/s ghi ở QD=8) với độ trễ phát sinh dưới 2%. Phân tích phần cứng chỉ ra sự đánh đổi rõ rệt giữa logic và bộ nhớ: hệ thống MKV-XTS tốn thêm 19,1% tài nguyên logic (50.232 LUT) so với cấu hình AES cơ bản, nhưng loại bỏ hoàn toàn sự phụ thuộc vào BRAM trong đường dẫn xử lý dữ liệu quan trọng.

Keywords— Network-Attached Storage (NAS); Field-Programmable Gate Array (FPGA); MKV block cipher; hardware encryption; data-at-rest protection; real-time processing; cryptographic IP core; SATA interface.

Từ khóa— Thiết bị lưu trữ mạng (NAS); FPGA; mã khối MKV; mã hóa phần cứng; bảo vệ dữ liệu tĩnh; thời gian thực; lõi IP mật mã; giao tiếp SATA.

I. INTRODUCTION

The explosion of digital data in recent years has established storage systems as a core component of modern information technology infrastructure. In this context, Network-Attached Storage (NAS) systems have been increasingly deployed due to their advantages in centralized storage, flexibility, and scalability. However, these benefits are accompanied by critical challenges regarding data security, particularly for data-at-rest. Threats such as unauthorized access, side-channel analysis, or physical cloning of storage devices can lead to severe information leakage if data is not protected by robust encryption mechanisms.

This manuscript was received on February 4, 2026. It was reviewed on April 15, 2026, revised on April 15, 2026 and accepted on May 5, 2026.

* Corresponding author

Currently, security solutions for NAS predominantly rely on software-based encryption deployed at the operating system, file system, or storage management layers. While offering flexibility, these approaches often result in significant performance degradation due to high CPU resource consumption. This creates I/O throughput bottlenecks and increases system latency, particularly in concurrent or random access scenarios. Consequently, this necessitates a dedicated parallel processing architecture capable of intercepting and processing data directly on the physical transmission path without burdening host resources.

To reconcile the conflicting demands of high performance and system customizability, FPGA-based inline encryption architectures have emerged as a promising solution. In this configuration, the FPGA acts as an intermediary bridge between the host system and the storage device, performing real-time encryption and decryption directly on the data stream. For block-based storage systems, the XTS mode is regarded as the standard and is recommended by international bodies due to its suitability for sector structures and high compatibility with traditional storage architectures.

However, it is important to note that the XTS standard only defines the overall encryption mode—including tweak generation and sector-level processing—without mandating the use of a specific block cipher. Consequently, the intrinsic characteristics of the encryption core, such as pipeline depth, hardware resource distribution, and data dependencies, can significantly impact the interaction between the encryption module and the storage datapath. To accurately assess real-world operational efficiency, especially when the system handles data-intensive I/O workloads, a comprehensive system-level evaluation approach is required. In this context, the encryption core is viewed not merely as a functional unit executing a fixed algorithm, but as a critical module that directly influences the entire storage datapath architecture.

While the AES currently stands as the predominant block cipher for secure storage, the cryptographic landscape is evolving. New algorithms are emerging to address specific constraints in hardware efficiency and to ensure

long-term security resilience. In this context, the MKV algorithm is introduced as a viable candidate for data-at-rest protection. It features a flexible design with dual block size options - 128-bit for transitional compatibility and 256-bit targeting post-quantum security assurance - supported by three variable key lengths. Crucially, the security margin of MKV has been rigorously verified against a comprehensive spectrum of modern cryptanalytic techniques, demonstrating strong resistance against differential, linear, boomerang, algebraic, integral, impossible differential, and related-key differential attacks.

To meet the high throughput demands of modern storage networks, recent FPGA-based AES-XTS implementations have heavily relied on core parallelization and deep pipelining [1, 2]. Similarly, industry applications increasingly deploy AES-XTS in high-performance computing (HPC) to secure data-at-rest without restricting data flow [3]. Yet, a common limitation of these previous studies is their tendency to evaluate cryptographic cores in isolation, separated from the actual storage protocol stack. This study bridges that gap by proposing a true system-level evaluation framework. By testing both the AES standard and the indigenous MKV cipher within the exact same XTS-based inline bridging architecture, we map the direct impact of core selection on end-to-end NAS performance. Ultimately, this empirical approach confirms MKV's practical viability for high-throughput storage environments.

This paper presents an empirical comparative study between MKV-XTS and AES-XTS configurations on an FPGA-based NAS platform. Unlike previous studies that evaluated algorithms only at the isolated core level, this research approaches the problem from a holistic storage system perspective. Specifically, the impact of cryptographic core selection is analyzed based on its actual interaction with the data transmission path and the inline processing architecture. To ensure objectivity, both cores are integrated into a unified XTS design framework, maintaining identical hardware conditions and test scenarios.

This paper makes the following specific contributions:

- Establishment of a standardized evaluation framework based on the XTS mode and inline FPGA architecture, facilitating the objective benchmarking of diverse block cipher algorithms within a NAS environment.

- Execution of a system-level quantitative comparison between MKV-XTS and AES-XTS configurations, focusing on hardware resource utilization, pipeline characteristics, and real-world performance.

- Analysis of NAS system performance behavior under the influence of distinct encryption cores operating within the same XTS mode.

- Extensive hardware synthesis and empirical performance evaluations demonstrate that the MKV-XTS implementation sustains multi-gigabit throughput (>490 MB/s), with a performance overhead of less than 2% relative to an unencrypted baseline. Furthermore, component-level analysis indicates that, although MKV incurs a 19.1% increase in logic utilization due to its combinational SubCells layer, it eliminates BRAM dependency within the critical datapath, resulting in a highly specialized and memory-efficient architecture for SoC integration.

Experimental results demonstrate that MKV is a competitive encryption solution, offering long-term viability for data-at-rest protection in NAS environments. It is particularly well-suited for applications demanding strict adherence to national standards, high security assurance, and hardware resource optimization.

II. EVALUATION FRAMEWORK ESTABLISHMENT

A. Securing data-at-rest in network-attached storage

In modern information technology infrastructures, Network-Attached Storage (NAS) systems play a pivotal role due to their capability to consolidate data and provide flexible access over TCP/IP networks. However, this centralization also renders NAS a prime target for large-scale data exfiltration attacks. While protection mechanisms for data-in-transit have been well-standardized, securing data-at-rest, specifically data residing on the physical layers of storage media, continues to face significant

challenges regarding both operational performance and long-term security assurance [1].

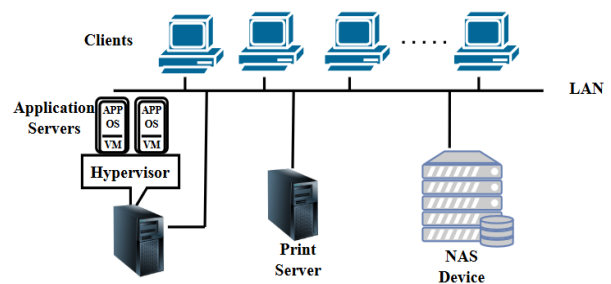


Figure 1. Typical NAS deployment architecture

The threat landscape for data-at-rest in NAS systems is characterized by a wide spectrum of vectors, notably:

- Physical attacks: Adversaries physically extract storage drives from the NAS appliance to mount and read data on a foreign system, effectively bypassing all network-level security perimeters.

- Digital forensics and cold-boot attacks: The use of specialized tools to recover residual data from memory or disk platters, even when the system is powered off or in a sleep state.

- Side-channel attacks (SCA): Measuring physical parameters such as power consumption or electromagnetic radiation during data access operations to infer cryptographic keys or plaintext content.

Due to inherent data redundancy mechanisms (e.g., RAID, snapshots) and periodic backup routines, if data-at-rest is not encrypted at the source, OS patches or firewalls become futile once the physical storage medium is compromised. To address this issue, three levels of encryption are typically implemented:

- (1) File-Level Encryption (FLE): Offers granular flexibility but is heavily dependent on software agents and host resources;

- (2) Full-Disk Encryption (FDE): Widely adopted, yet its protection is typically compromised once the operating system is booted and the volume is mounted;

- (3) Block-Level Encryption: Operates directly on fixed-size data blocks independently of the file structure.

Within the domain of high-performance NAS environments, hardware-accelerated block-

level encryption emerges as the paramount solution. This approach operates independently of the operating system, enabling real-time processing capabilities while effectively preventing direct physical access attacks. Integrating the encryption module into programmable logic circuits, such as FPGAs or SoCs, not only ensures high throughput performance but also allows for secure control over the entire data processing chain.

Currently, the field of hardware encryption faces pivotal technological challenges that necessitate significant design innovations:

- Resource constraints: Despite being the ubiquitous standard, high-performance AES implementations demand substantial logic resources (primarily LUTs) and complex pipeline depths. This imposes significant optimization difficulties on low-cost FPGA families or energy-constrained NAS devices.

- Technological diversification and sovereignty: Exclusive reliance on a single international algorithm poses potential long-term security risks. The development and integration of indigenous encryption cores (such as MKV) not only ensure technological autonomy but also ensure compliance with specific national security standards.

- Post-quantum resilience: The dawn of the quantum era jeopardizes contemporary cryptographic primitives. It is imperative that algorithms demonstrate resilience against quantum vectors, specifically regarding the key space exhaustion capabilities of Grover's algorithm.

Against this backdrop, the MKV algorithm, distinguished by its enhanced SPN architecture, stands out as a viable contender. Nevertheless, validating its efficacy requires empirical benchmarking against AES within a tangible hardware environment. Therefore, this research proposes a highly adaptive framework, paving the way for the seamless integration of heterogeneous cryptographic algorithms into high-performance NAS architectures.

B. FPGA-based inline hardware encryption evaluation framework

To perform quantitative analyses regarding the differences between cryptographic algorithms,

this study employs an FPGA-based inline hardware encryption framework. This framework serves a dual purpose: acting as a practical implementation solution while simultaneously functioning as a neutral evaluation environment. This design allows us to isolate and focus solely on assessing the impact of different block cipher cores (MKV and AES) on the overall storage system performance.

Within this model, the FPGA operates as a transparent bridge between the host controller and the storage device (SSD). It facilitates real-time data encryption and decryption directly on the SATA transmission stream, without requiring any modifications to the operating system, NAS software, or the storage device's firmware.

Figure 2 illustrates the overarching architecture of the evaluation framework, designed to objectively assess the impact of cryptographic algorithms on system performance. Within this architecture, the complex protocol layers (SATA Host/Device Controllers) and data transport logic constitute a fixed evaluation infrastructure, serving to maintain a stable data transmission link between the host and the storage device.

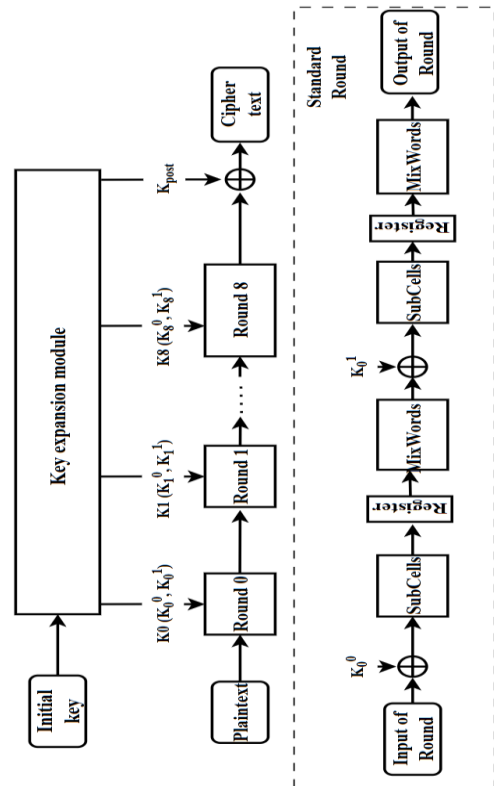


Figure 2. Modular framework for comparative evaluation of block ciphers in NAS environments

The core component and focal point of the framework is the Modular XTS Cipher Engine. This block is designed not merely for encryption execution but specifically for comparative benchmarking, incorporating the following features:

- Unified core infrastructure: Within the cipher engine, two cryptographic IP cores are instantiated in parallel on the same datapath:

- + AES-256 Core (Reference): Acts as the reference baseline, representing the performance of prevailing industry standards.

- + MKV Core (Evaluation): Represents the indigenous block cipher algorithm under evaluation.

Both encryption cores share identical input vectors, including plaintext, key, and the tweak value derived from the sector address. The sharing of these input signals ensures that all data boundary conditions are strictly uniform for both algorithms.

- Synthesis-time selection mechanism: Instead of employing a dynamic multiplexer (MUX) that would instantiate both cores simultaneously and consume redundant FPGA area, the architecture utilizes a parameter-driven configuration mechanism. During hardware synthesis, only one of the two algorithm types is selectively instantiated to accurately reflect its standalone hardware footprint:

- + When configured for AES: Only the AES cores are synthesized into the data stream.

- + When configured for MKV: Only the MKV cores are synthesized into the data stream.

Performance evaluation objectivity: By placing both cores in a parallel configuration within the same functional block, the timing constraints and routing delays from the SATA controllers remain identical for both ciphers. Consequently, any measured variations in end-to-end throughput or latency accurately reflect the intrinsic architectural differences (e.g., pipeline depth, critical path) between MKV and AES, effectively eliminating extraneous noise from the protocol environment.

The design of the evaluation framework, featuring parallel core architecture and a transparent data selection mechanism, establishes a

standardized experimental environment, ensuring maximum objectivity for the comparative assessment. By fixing environmental variables - such as the SATA protocol and buffer management - the system ensures that all measured discrepancies are directly attributable to the architectural characteristics of the encryption algorithms rather than external factors. This provides a critical empirical foundation for the reliability and accuracy of the resource and performance analyses presented in the subsequent experimental results section.

C. Unified XTS architecture for cross-algorithm evaluation

To establish an objective and consistent evaluation framework for diverse encryption algorithms, this study standardizes the operating mode to XTS, adhering to the IEEE Std 1619 [10] specification. Within the proposed architecture, the XTS layer functions as the execution environment, governing tweak key generation and sector-level data management. Conversely, the underlying central block cipher (AES or MKV) is abstracted to focus exclusively on the transformation of 128-bit data blocks.

The selection of XTS is particularly advantageous for secure NAS environments, as it effectively mitigates the inherent limitations of traditional chaining modes (such as CBC), which are constrained by sequential processing dependencies and lack parallelization capabilities. Crucially, XTS facilitates random access to data within a sector without dependencies on preceding blocks. Furthermore, it robustly defends against replay attacks and ciphertext manipulation by cryptographically binding the encryption process to specific Logical Block Addresses (LBAs).

Mathematically, XTS operates as an extension mechanism that integrates a position-dependent "tweak" value derived from the LBA. This ensures that each data block is encrypted independently, guaranteeing that identical plaintext blocks produce distinct ciphertext outputs when stored at different physical locations.

In detail, for a given sector consisting of m 128-bit blocks, the transformation of each plaintext block P_j to its corresponding ciphertext C_j proceeds via two main stages:

- Step 1: Tweak Generation. The tweak value T_j is computed from the sector's Logical Block Address (LBA) utilizing a secondary tweak key K_2 :

$$T_j = E_{K_2}(LBA) \otimes \alpha^j$$

Where $\otimes$ denotes multiplication in the Galois Field $GF(2^{128})$ modulo the irreducible polynomial $x^{128} + x^7 + x^2 + x + 1$, and α corresponds to the primitive element of the field.

Multiplication by successive powers of α enables the derivation of a distinct Tweak value for every block within a sector based on a singular LBA seed. This process effectively enforces the property of spatial variability.

- Step 2: Data encryption with the primary key K_1 :

$$C_j = E_{K_1}(P_j \oplus T_j) \oplus T_j$$

Here, $E_K(\cdot)$ denotes the underlying block cipher function, and $\oplus$ represents the fundamental bitwise XOR operation. This mechanism ensures that identical plaintext blocks stored at different physical locations yield distinct ciphertexts, thereby enhancing resistance against manipulation and mitigating pattern-based attacks.

When deploying XTS in storage environments, a critical challenge arises regarding the stringent requirements for high throughput and low latency, as data must be processed directly within the datapath. A sequential implementation of the aforementioned operations would inevitably create significant performance bottlenecks, particularly when each sector contains multiple 128-bit blocks requiring continuous processing. Consequently, the XTS architecture proposed in this study is organized based on pipelining and parallelization principles:

- Tweak generation path: An auxiliary encryption core is utilized to encrypt the LBA value using key K_2 to derive the initial tweak T_0 . Subsequent tweak values T_j are generated sequentially via hardware-accelerated Galois multiplication, ensuring a continuous supply of tweak values for each data block without stalling the data stream.

- Data processing path: This constitutes the primary datapath housing the central encryption core. Input sector data is XORed with the

corresponding tweak prior to entering the encryption core, and the output is subsequently XORed again to produce the final ciphertext. The pipelined mechanism allows each block to be processed continuously without waiting for the completion of the entire sector, thereby achieving significantly higher throughput compared to sequential processing.

This parallelized and pipelined XTS implementation - characteristic of high-performance FPGA designs - enables multi-Gbps throughput while maintaining reasonable resource utilization costs.

A paramount architectural advantage of the XTS mode is the strict decoupling between the control logic - encompassing tweak generation and XOR operations - and the cryptographic execution core. By adhering to this design principle, infrastructural components such as the pipeline structure, SATA interface, and tweak management mechanism become fixed elements, operating independently of the specific internal encryption algorithm. This facilitates the hot-swappability of encryption cores without compromising the integrity of the inline processing architecture. This serves as the foundational premise enabling this study to conduct a comparative benchmarking of MKV and AES on a unified hardware platform, thereby eliminating errors or biases stemming from integration environment discrepancies.

D. NAS integration constraints and architectural motivation

The architectural strategies adopted for both the AES and MKV encryption cores in this study are not arbitrary; rather, they are rigorously derived from the integration constraints of an inline NAS environment. As the cryptographic module operates as a transparent bridge within the physical datapath, it must be tightly aligned with the storage protocol stack, leading to several fundamental hardware requirements:

SATA datapath timing and throughput:

Modern NAS systems based on SATA Gen3 interfaces require sustained data rates of up to 6 Gbps. To ensure that the encryption layer does not become a throughput bottleneck, the cryptographic core must support a data

processing rate that matches or exceeds the physical link capacity. As a result, compact iterative architectures-where a single combinational unit is reused over multiple clock cycles - are unsuitable. This stringent throughput requirement necessitates the use of fully unrolled and deeply pipelined architectures for both AES and MKV implementations.

Inline latency constraints and FIFO buffering:

In a transparent inline bridging configuration, the FPGA intercepts the data stream between the SATA host and device controllers. Excessive or non-deterministic processing latency can lead to rapid overflow of intermediate FIFO buffers, resulting in data loss or protocol timeouts. Therefore, the design must guarantee fixed and predictable latency. The deep pipeline structure of the AES core, along with the fine-grained pipelining strategy employed in the MKV core, ensures a continuous and deterministic dataflow, maintaining FIFO utilization within safe operational limits.

Sector-aligned processing requirements:

Storage systems operate on fixed-size sectors (typically 512 bytes or 4 KB). Under XTS mode, tweak values must be generated and applied synchronously with the corresponding data blocks within each sector. The hardware architecture must handle sector boundaries seamlessly without disrupting the pipelined datapath. This requirement enforces tight integration between the tweak generation logic and the primary encryption pipeline, ensuring that sector transitions do not introduce additional clock cycle stalls. In summary, the constraints imposed by inline NAS integration render low-area, high-latency iterative cipher architectures impractical. The combined requirements of high throughput, deterministic latency, and sector-aware processing fundamentally justify the adoption of fully unrolled, high-performance pipelined architectures, as analyzed in Section III.

III. INTERNAL ARCHITECTURE OF ENCRYPTION CORES

While the fundamental algorithmic structures of AES and MKV differ significantly, this section focuses primarily on the specific architectural properties - such as S-box realization and diffusion layer complexity - that directly dictate their pipeline

design and hardware resource footprint within the proposed XTS integration.

A. AES-256 encryption core architecture

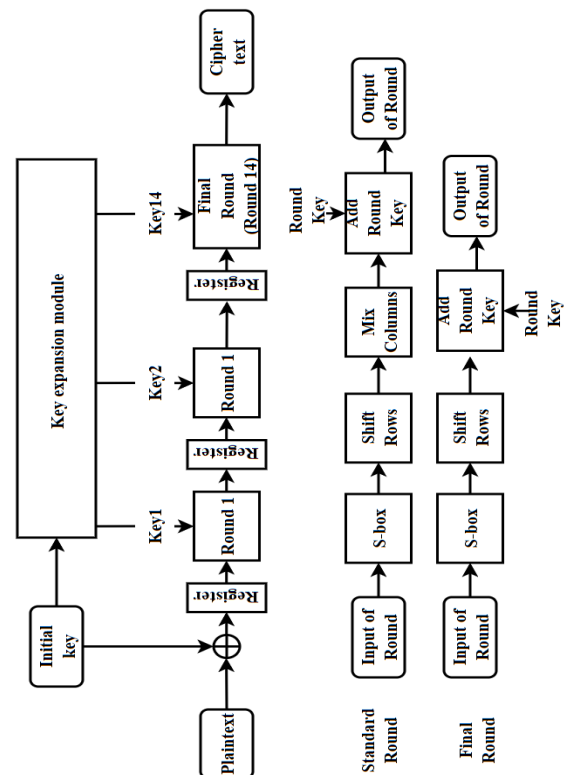


Figure 3. Pipelined AES-256 core implementation

The Advanced Encryption Standard (AES) operates on fixed 128-bit data blocks and is structured as a Substitution–Permutation Network (SPN). For the 256-bit key configuration (AES-256), the algorithm consists of 14 processing rounds. To meet the high-throughput requirements of inline Network Attached Storage (NAS) systems, the AES core is typically implemented using a fully unrolled pipelined architecture, in which each round is mapped to a dedicated pipeline stage.

The hardware resource utilization of AES on FPGA platforms is primarily dominated by its non-linear substitution and linear diffusion operations. Specifically, the SubBytes transformation employs 8-bit S-boxes, which are commonly implemented using Look-Up Tables (LUTs) to perform arithmetic in $GF(2^8)$, thereby consuming a significant portion of the logic fabric. In addition, the MixColumns operation involves matrix multiplications over the finite field, leading to increased combinational logic

complexity. As a result, achieving sustained, stall-free SATA data throughput with AES-256 requires a substantial number of LUTs as well as pipeline registers (Figure 3).

B. MKV Block Cipher Algorithm and Design Principles

The MKV block cipher is constructed using an extended SPN combined with a Feistel-Like Construction (FLC) [15]. Crucially, MKV's internal cryptographic layers are intrinsically optimized for hardware mapping. The SubCells layer employs an 8-bit S-box tailored for butterfly-style bit-slicing [17]. This design maps naturally to the native logic cells of the FPGA fabric, drastically reducing the reliance on LUTs compared to traditional AES implementations. Furthermore, the MixWords diffusion layer utilizes an optimized Maximum Distance Separable (MDS) matrix based on simple cyclic shifts and bitwise XOR operations, which minimizes routing complexity.

TABLE I. ARCHITECTURAL COMPARISON OF AES AND MKV FOCUSING ON HARDWARE IMPLEMENTATION IMPLICATIONS

Feature	AES-256 (Ref.)	MKV-128 (Eval.)	HW Implementation Impact
Core Structure	Standard SPN	Extended SPN with FLC [19]	Dictates data routing & pipeline register placement
Block / Key Size	128b block / 256b key	128b block / 256b key	Ensures identical datapath widths & XTS tweak integration
Processing Rounds	14 rounds (14 pipeline stages)	9 rounds (for equivalent security)	Determines pipeline depth, critical path & total latency
S-box	8-bit, LUT-based in $GF(2^8)$	8-bit, butterfly-style bit-sliced	Primary driver of LUT consumption on FPGA fabric
Diffusion Layer	MixColumns (MDS matrix)	Optimized MDS (rotation & mult.)	Affects logic density & routing delay per stage
Key Schedule	On-the-fly, linear derivation	Iterative, non-linear derivation	Impacts memory (BRAM) & intermediate register (FF) usage

To sustain the continuous multi-gigabit throughput required by the XTS datapath, the MKV hardware implementation employs a fine-grained intra-round pipelining strategy. Specifically, intermediate pipeline registers are strategically inserted immediately following every SubCells layer. This structural placement effectively isolates the combinational logic of the non-linear substitution from the subsequent MixWords diffusion layer. While this design choice deepens the overall pipeline structure and inherently increases Flip-Flop (FF) utilization to store intermediate states, it is a necessary architectural adaptation. It allows the MKV core to meet the strict throughput constraints of the SATA interface while capitalizing on its bit-sliced architecture to maintain a highly optimized LUT resource profile.

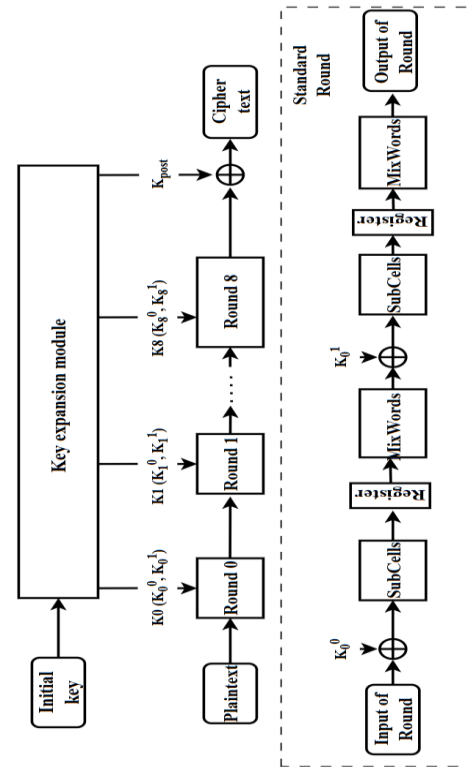


Figure 4. Implementation architecture of the MKV-128 core with a 256-bit key

The FPGA-based pipelined implementation of the MKV-128 encryption core, configured with a 256-bit key, is architected as Figure 4.

Internal operations of the MKV core: Each processing stage (round) of the MKV algorithm executes the following cryptographic transformations [15]:

- SubCells: A non-linear substitution layer where 8-bit input segments are replaced using a dedicated S-box.

- MixWords: A linear diffusion layer. For each input state, MixWords updates individual sub-states through a composite linear transformation consisting of cyclic left shifts (rotations) and matrix multiplication.

- XWords: A mixing operation that updates the state by performing bitwise XOR addition between sub-states to generate the final output value.

In terms of security assurance, it is essential to recognize the differing levels of analytical maturity between the two ciphers. AES has undergone more than two decades of extensive, independent, and global cryptanalysis, demonstrating strong resistance against all known classes of attacks and establishing itself as the de facto standard for symmetric encryption. In contrast, MKV is a relatively recent cryptographic design. The security properties attributed to MKV in this study are based on currently available analyses and the design claims provided by its developers, including resistance to contemporary attack vectors such as differential and algebraic cryptanalysis. However, MKV has not yet achieved the same level of long-term, widespread validation as AES. Therefore, MKV can be regarded as a promising alternative that offers theoretical robustness and hardware efficiency, whereas AES continues to serve as the benchmark for thoroughly validated and widely trusted security. The following section presents a comparative analysis of the pipelined AES-256 core [24] and the pipelined MKV core, implemented on the identical hardware platform. The comparison focuses on key performance indicators as shown in Table II:

TABLE II. COMPARISON OF PIPELINED AES AND MKV ENCRYPTION CORES BASED ON PERFORMANCE METRICS

Measurement index	AES-256 (Pipelined)	MKV (Pipelined)
Pipeline depth	14	9
Latency (Clock Cycles)	16	20
Max Frequency MHz	235	220

Performance analysis and observations:

- Pipeline structure: AES-256 requires 14 processing rounds, with each round mapped to a single pipeline stage. In contrast, MKV, utilizing a Feistel-Like Construction (FLC) and optimized S-boxes, achieves equivalent cryptographic security strength with fewer processing stages.

- Latency analysis: The processing latency of each stage in the MKV architecture corresponds to the number of registers inserted after each SubCells block. These registers serve to store intermediate data before it is forwarded to the MixWords operation. Since each stage performs the SubCells transformation twice, the processing latency per stage in MKV is two clock cycles, resulting in a total latency of 20 clock cycles. Thus, the timing difference between the two cores is minimal, despite the significant reduction in the number of logical rounds.

- Frequency and applicability: The Maximum Operating Frequency $F_{\max}$ shows negligible difference between the MKV and AES-256 cores. With such frequency performance, deploying MKV for data encryption in SATA Gen 3 storage applications on FPGAs is highly suitable and compatible with the processing speed of standard IP Cores.

Although a 256-bit block variant of MKV is available, we deployed MKV-128 with a 256-bit key to align perfectly with the native structure of AES-256. This structural alignment eliminates hardware discrepancies; it guarantees that the datapath widths, XTS tweak computations, and sector-level processing logic are uniform across both evaluation cores, yielding a highly accurate performance baseline.

IV. FPGA IMPLEMENTATION AND PERFORMANCE EVALUATION

A. General architecture of the evaluation model

To validate the feasibility and efficacy of the proposed architecture, the system was physically implemented on an actual FPGA hardware platform. The primary objective of the experimental phase is to conduct a quantitative comparison between the standard AES-256 core and the proposed MKV core, focusing on three key metrics: hardware resource utilization, data transmission throughput, and system latency.

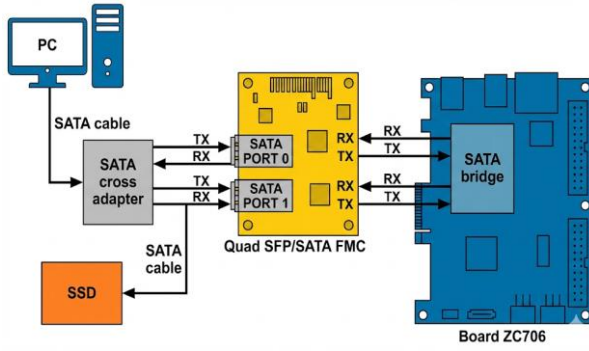


Figure 5. Overall system architecture

The system was deployed on the Xilinx Zynq-7000 SoC ZC706 Evaluation Kit. This platform is particularly well-suited for storage applications due to its integration of ARM processors and high-density programmable logic on a single chip. Interfacing with storage drives is established via a Quad SFP/SATA FMC expansion module connected through the FMC HPC (High Pin Count) port, fully supporting the SATA Gen 3 standard.

- Testbed configuration: The experimental apparatus is configured with the following specifications:

+ Host machine: Intel Core i5-6500 CPU @ 3.20 GHz, 8 GB RAM, running Ubuntu 20.04 (for development) and Windows 10 (for benchmarking).

+ Storage Device: TeamGroup CX2 256 GB SSD, 2.5-inch SATA III standard.

+ Synthesis and implementation tool: Xilinx Vivado Design Suite 2022.2.

- Instrumentation and metrology:

+ CrystalDiskMark 8.0.5: For quantitative throughput measurement.

+ HxD hex editor: For low-level sector data inspection and verification.

+ Xilinx power estimator (XPE): For power consumption estimation.

- Functional verification and data integrity ensuring the integrity and correctness of cryptographic operations is a mandatory prerequisite prior to performance benchmarking. The verification process involves two phases:

+ Sector-level verification: A known data pattern (e.g., repeating 0x11 bytes) is written to Sector 0 via the FPGA system. Subsequently, the

SSD is connected directly to a PC (bypassing the FPGA). The data read from the drive appears as random ciphertext/noise, confirming that encryption is actively occurring on the storage path.

+ Cross-validation and algorithmic compliance: Data decrypted by the FPGA hardware is verified to match the original plaintext bit-for-bit. Furthermore, standard test vectors for both MKV and AES are utilized to validate that the hardware output strictly adheres to the official algorithmic specifications.

Prior to the resource utilization and performance analysis, both encryption cores were successfully integrated into the storage hierarchy, with their functional correctness and data integrity fully validated.

B. FPGA resource consumption results

To provide a comprehensive evaluation of FPGA resource consumption, two distinct secure data storage systems - utilizing the AES-256 and MKV-128 (with a 256-bit key) encryption cores, respectively - were synthesized and implemented on the Xilinx ZC706 evaluation platform. To ensure the highest degree of fairness and statistical validity in the comparative analysis, both systems were developed using identical hardware configurations, including the same bus architectures and memory interfaces. Furthermore, a unified set of synthesis and implementation tools, specifically the Vivado Design Suite, was employed with consistent optimization constraints. The resulting post-implementation resource utilization metrics, encompassing LUTs, registers, and memory blocks for each core configuration, are detailed and compared in the following Table III.

TABLE III. HARDWARE RESOURCE UTILIZATION OF THE COMPLETE XTS CIPHER ENGINES

Resources	AES	AES utilizations (%)	MKV	MKV utilizations (%)
LUT	36,611	16.74%	45,244	20.69%
FF	15,459	3.53%	16,992	3.88%
BRAM	0	0%	1	0.18%

To accurately assess the intrinsic hardware efficiency of the encryption algorithms independent of the SATA bridge protocol overhead, the core cryptographic modules were isolated. Table III presents the isolated hardware resource utilization of the fundamental

encryption, decryption, and key generation modules, while Table IV details the estimated component-level resource distribution derived from the synthesis hierarchy.

It is acknowledged that prior studies have proposed highly area-efficient AES implementations. For example, Ali et al. [28] and Khalid et al. [29] introduced compact architectures based on folded or reduced-width datapaths, while Zhang et al. [30] proposed S-box optimizations using Composite Field Arithmetic (CFA) to minimize logic complexity. However, these approaches inherently rely on iterative processing or introduce longer critical paths, making them unsuitable for sustaining continuous multi-gigabit data rates. To overcome these limitations, our design adopts a high-performance XTS architectural methodology akin to that presented in [24], which necessitates the instantiation of three concurrent cryptographic modules: two parallel encryption cores (dedicated to data writing and consecutive tweak generation) combined with one decryption core (dedicated to data reading). Furthermore, the isolated AES and MKV cores in this work adopt fully unrolled, deeply pipelined architectures, ensuring deterministic, high-throughput operation. Consequently, the resource utilization reported in Table III represents a realistic and mandatory baseline for line-rate NAS performance without pipeline stalls.

TABLE IV. COMPONENT-LEVEL RESOURCE BREAKDOWN OF INDIVIDUAL CRYPTOGRAPHIC MODULES

Cryptographic Module	AES-256 Core (LUTs/FFs)	MKV-128 Core (LUTs/FFs)
Single Encryption Core (Data Write / Tweak)	10,009/ 4,098	13,315/ 4,738
Key Expansion / Schedule	2,324/ 2,376	1,053/ 917 (+ 1 BRAM)
XTS Control Logic & Routing	17,812/ 4,887	26,078/ 6,599

Within this design framework, the MKV-128 encryption core consumes 13,315 LUTs, exceeding the AES-256 encryption core (10,009 LUTs). As shown in Table V, the SubCells (S-box) layer accounts for a dominant portion of this usage (74.0% of LUTs). This reflects a deliberate architectural choice to implement the non-linear

layer entirely using combinational logic, thereby avoiding reliance on memory-based structures. This approach removes BRAM dependency from the high-speed datapath and ensures consistent throughput.

Additionally, Table V indicates that the MKV MixWords diffusion layer is more logic-efficient than AES MixColumns, accounting for only 11.0% of LUT usage versus 24.0% in AES, due to its optimized cyclic-shift MDS structure. The MKV key schedule module is also compact, utilizing 1,053 LUTs and a single BRAM block to manage its non-linear iterative process, achieving an effective balance between logic and memory resources.

TABLE V. ESTIMATED COMPONENT-LEVEL RESOURCE DISTRIBUTION WITHIN THE ENCRYPTION CORES

Cryptographic Component	AES-256 Core (LUTs/FFs)	MKV-128 Core (LUTs/FFs)
Non-linear Layer (S-box/SubCells)	~62.0%/ ~28.0%	~74.0%/ ~41.0%
Diffusion Layer (MixCol/MixWords)	~24.0%/ ~12.0%	~11.0%/ ~8.0%
Key Schedule Generation	~8.0%/ ~15.0%	~9.0%/ ~18.0%
Pipeline Control & Routing Logic	~6.0%/ ~45.0%	~6.0%/ ~33.0%

To further elucidate the distribution of logic resources and support the architectural design decisions, Table V presents the estimated component-level resource breakdown within the integrated XTS engines, as derived from the synthesis hierarchy:

Non-linear Layer Overhead: The MKV SubCells layer dominates LUT and FF utilization due to its fully combinational implementation and fine-grained pipelining. This design eliminates BRAM usage in the datapath and ensures high-throughput operation, at the cost of increased logic resources.

Diffusion Layer Efficiency: In contrast, the MKV MixWords layer is significantly more efficient than AES MixColumns, as it relies on simple shift and XOR operations rather than complex GF(2⁸) matrix multiplications, reducing both logic usage and routing delay.

Key Schedule and Pipeline Control: Both designs maintain moderate key schedule overhead, but the fully unrolled architectures require additional resources for pipeline control and state propagation to sustain continuous 6 Gbps throughput.

In summary, MKV achieves high efficiency in its diffusion layer, while accepting increased LUT usage in the SubCells layer as a trade-off for a memory-free, high-throughput datapath.

TABLE VI. TOTAL HARDWARE RESOURCE UTILIZATION OF THE INTEGRATED SECURE NAS BRIDGING SYSTEMS

Resources	AES	AES utilizations (%)	MKV	MKV utilizations (%)
LUT	42,171	19.29	50,232	22.98
FF	21,861	5.00	23,397	5.33
BRAM	33	6.06	49	8.99
LUTRAM	246	0.35	374	0.53

When transitioning from isolated cryptographic cores to the complete inline bridging system, overall hardware utilization increases to accommodate both the XTS engines and the supporting storage infrastructure. As shown in Table VI, the reported metrics include not only the dual-core XTS modules (encryption, decryption, and tweak generation) but also the SATA PHY, Link/Transport layers, AXI interconnects, and intermediate FIFO buffers.

The AES-XTS system establishes a baseline with 42,171 LUTs and 33 BRAMs. In comparison, the MKV-XTS system utilizes 50,232 LUTs (+19.1%) and 23,397 FFs (+7.0%), reflecting the additional logic required by the MKV architecture - particularly within the combinational SubCells layer - to maintain high-throughput, memory-independent datapath execution.

Additionally, the MKV-XTS system consumes 49 BRAMs, compared to 33 BRAMs in the AES design. Given that the isolated MKV core requires only minimal BRAM for key scheduling, this increase is primarily attributed to system-level buffering, including FIFO structures and AXI clock-domain crossing. These additional memory resources are necessary to sustain

continuous 6 Gbps dataflow and ensure reliable synchronization without data loss.

Overall, the results demonstrate that MKV is well-suited for high-speed storage security applications, achieving multi-gigabit throughput by leveraging FPGA logic resources, and offering a competitive, hardware-oriented alternative to AES.

C. Storage performance evaluation

To comprehensively evaluate the impact of the encryption process on overall system performance, throughput measurements were conducted using the CrystalDiskMark benchmarking utility. To reflect a representative NAS workload profile, the evaluation methodology incorporates multiple access patterns and varying queue depths (QD). Specifically, large file transfer scenarios are modeled using 1 MB sequential access at QD = 8 and QD = 1, while small-file, concurrent workloads are represented by 4 KB random access at QD = 32 and QD = 1, all executed under a single-threaded configuration (T = 1). The results are compared across three scenarios: (1) No Encryption (Baseline), (2) System with AES-256, and (3) System with MKV.

Discussion and evaluation:

Under 4 KB random I/O workloads, both encryption cores demonstrate stable performance across both shallow (QD = 1) and deep (QD = 32) queue depths. Notably, the random write throughput remains consistently competitive, indicating that the pipelined encryption architectures effectively handle concurrent and irregular write operations without introducing significant latency to the SSD controller.

The empirical results presented in Table VII validate the efficacy of the proposed inline encryption architecture under realistic storage workloads. The observed sequential throughputs for both cryptographic cores converge near the physical saturation point of the test SSD, which operates at a baseline of 499.12 MB/s for reads. Specifically, the MKV-XTS configuration achieves 490.28 MB/s for sequential reads and 433.15 MB/s for sequential writes at a queue depth of 8 (QD=8). This confirms that

integrating the pipelined encryption layer introduces acceptable performance overhead to the data path, ensuring that the FPGA bridge does not constitute a bottleneck for SATA storage interfaces. Relative to the unencrypted baseline, the MKV-XTS configuration incurs a minimal sequential throughput penalty of only 1.77% for reading and 1.89% for writing. This overhead is highly comparable to the 1.27% drop observed with the AES-XTS implementation, indicating that the MKV-XTS pipeline functions effectively without imposing excessive propagation delay on the continuous SATA stream. Furthermore, under demanding random 4KB I/O workloads, both cores exhibit sustained stability across both shallow (QD=1) and deep (QD=32) queues. The MKV-XTS random write performance reaches 212.81 MB/s (QD=32), confirming its capability to handle complex, concurrent read and write requests efficiently.

TABLE VII. COMPARISON OF SEQUENTIAL AND RANDOM READ/WRITE SPEEDS: BASELINE VS. AES AND MKV

Block Cipher Configuration	Seq 1M Q8T1 Read/Write (MB/s)	Seq 1M Q1T1 Read/Write (MB/s)	RND 4K Q32T1 Read/Write (MB/s)	RND 4K Q1T1 Read/Write (MB/s)
Baseline (No encryption)	499.12/ 441.50	475.29/ 422.34	175.20/ 216.87	37.15/ 91.56
AES-256-XTS	492.76/ 436.45	468.32/ 415.82	172.10/ 212.90	35.88/ 88.34
MKV-XTS	490.28/ 433.15	465.18/ 412.56	170.45/ 212.81	35.12/ 87.20

A holistic evaluation of both hardware resource consumption and storage performance reveals the practical feasibility of the MKV algorithm. Foremost, the MKV-XTS satisfies modern NAS throughput demands, demonstrating performance levels comparable to the industry-standard AES-XTS in both sequential file transfers and random concurrent I/O operations. In terms of hardware utilization, MKV presents a distinct architectural trade-off. By utilizing combinational assign logic for the SubCells layer, the design eliminates BRAM dependency within the high-speed data processing path, albeit at the cost of a higher logic density. This approach preserves the FPGA's embedded memory strictly for protocol FIFO

buffers and key scheduling. Additionally, the Feistel-Like Construction (FLC) and MixWords layer of MKV provide a practical and efficient logic footprint during the decryption process.

Operating under the IEEE 1619 XTS standard, both implementations ensure strong data-at-rest confidentiality at the sector level. The architectural design guarantees that even partial physical extraction of the SSD yields no usable plaintext, while system resilience is maintained through atomic sector operations and automated key reinitialization upon reset. Ultimately, this analysis establishes MKV-XTS as a viable and promising candidate for secure storage cores. It offers a functional alternative to AES for specialized infrastructure where sovereign cryptographic control is prioritized.

V. CONCLUSION

This study details the implementation and empirical analysis of an FPGA-based inline encryption architecture tailored for NAS ecosystems. Utilizing a transparent bridging mechanism within the SATA datapath, the proposed design effectively mitigates data-at-rest security risks with negligible latency penalties, offering a rigorous comparative analysis between the industry-standard AES-256 and the indigenous MKV cipher in XTS mode.

Line-rate performance: Both ciphers achieve throughput saturation close to the SATA Gen3 limit (~490 MB/s read and ~433 MB/s write). Under intensive workloads, including 4 KB random I/O at multiple queue depths, the MKV-XTS system introduces less than 2% performance overhead compared to the unencrypted baseline, demonstrating its suitability for real-time, transparent storage encryption.

Logic-memory architectural trade-off: Component-level analysis highlights a distinct hardware profile for MKV. To sustain 6 Gbps throughput without pipeline stalls, the SubCells layer is implemented using fully combinational logic. This approach inherently increases logic utilization (+19.1% LUTs compared to AES) but effectively guarantees a memory-free datapath, ensuring that embedded memory is preserved for system-level protocol buffers.

Additionally, the MKV decryption core maintains a compact and practical logic footprint. Consequently, while AES-XTS remains the established global standard for interoperability, MKV-XTS serves as a viable, hardware-aware alternative for specialized high-throughput storage infrastructure.

Strategic implications and future work: The results position MKV as a strong candidate for cryptographic sovereignty, offering structural flexibility and potential resilience to emerging threats. This work provides a practical bridge between cryptographic design and system-level deployment. Future efforts will focus on integrating data integrity mechanisms (e.g., HMAC or GCM) and extending the architecture to next-generation high-speed interfaces such as NVMe.

REFERENCES

- [1] S. An and S. C. Seo, "Designing a New XTS-AES Parallel Optimization Implementation Technique for Fast File Encryption," *IEEE Access*, vol. 10, 2022.
- [2] A. Alhomoud, "Real Time FPGA Implementation of a High Speed for Video Encryption and Decryption System with High Level Synthesis Tools," *IJACSA*, vol. 15, no. 1, 2024.
- [3] A. Alhomoud, "Real Time FPGA Implementation of a High Speed for Video Encryption and Decryption System with High Level Synthesis Tools," *IJACSA*, vol. 15, no. 1, 2024.
- [4] National Institute of Standards and Technology (NIST), "FIPS PUB 197: Advanced Encryption Standard (AES)" Nov. 2001.
- [5] S. Gueron, "Intel Advanced Encryption Standard (AES) Instructions Set," *Intel White Paper*, 2010.
- [6] T. Good and M. Benaissa, "AES on FPGA-from the fastest to the smallest," in *CHES, Springer*, pp. 427–440, 2005.
- [7] O. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *CRYPTO'99*, pp. 388–397, 1999.
- [8] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *STOC'96, ACM*, pp. 212–219, 1996.
- [9] Cuong Nguyen, Anh Nguyen, Phong Trieu, Long Nguyen, and Lai Tran, "Analysis of a new practically secure SPN-based scheme in the Luby-Rackoff model" in *The 9th International Conference on Future Data and Security Engineering*, 2022.
- [10] T. S. Nam, N. V. Long, N. B. Cuong, "An Optimized Bit-Slice Implementation of Secure 8-Bit Sbox," in *IEEE KSE*, 2023.
- [11] T. S. Nam et al., "A High Throughput, Low Latency 105 Gbps Four-Pipeline Stage AES", *Journal of Information Security Science and Technology*, vol 1. no. 21, pp.58-66, 2024. DOI: 10.54654/isj.v1i21.1029.
- [12] N. B. Cuong et al., "FLC: a new secure and practical SPN-based scheme," in *NAFOSTED NICS, IEEE*, 2022.
- [13] Internal Evaluation Report of MKV FPGA Testbench, Government Cipher Committee, 2023.
- [14] H. V. Thuc and N. B. Cuong, "Selected Results in the Development of Civil Block Cipher Standards," *Vietnam Cipher Journal*, no. 4, pp. 9 -12, 2022.
- [15] TCVN 14263-2024: Information technology – Security techniques – MKV block cipher algorithm.
- [16] N. Q. Toan and N. B. Cuong, "Development of "Make in Vietnam" Civil Cryptography Standards", *Journal of Information Security*, no. 2, 2023.
- [17] T. S. Nam, N. V. Long, and N. B. Cuong, "Bit-Slice Implementation of Secure 8-Bit Sbox," in *IEEE KSE*, 2023.
- [18] Draft Technical Specification of MKV, Government Cipher Committee, 2023.
- [19] Luong, T. T., "Building the dynamic diffusion layer for SPN block ciphers based on direct exponent and scalar multiplication", *Journal of Science and Technology on Information Security*, vol 1, no 15, pp. 38-45, 2022.
- [20] T. S. Nam, N. V. Long, N. B. Cuong, "An efficient and secure linear diffusion layer for 256-bit block cipher based on FLC structure," *Journal of Science and Technology on Information Security*, vol. 2, no. 16, pp. 31-38, 2022. DOI: 10.54654/isj.v1i16.920.
- [21] N. B. Cuong et al., "FLC: a secure and practical SPN-based scheme," in *NAFOSTED NICS, IEEE*, 2022.
- [22] Internal Report: Evaluation of MKV's Cryptanalysis Resistance, Government Cipher Committee, 2023.
- [23] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," in *Proc. of the 28th ACM Symposium on Theory of Computing*, 1996.
- [24] Tran Van Khanh, Phan Van Ky, Vu Van Viet, "FPGA-Based inline encryption bridge using AES-XTS for storage systems", *Journal of Science and Technology on Information*

Security, vol 3, no 26. pp 5-17, 2025. DOI: 10.54654/isj.v2i25.1141.

- [25] S. Ghosh et al., "Hardware Security in Storage Systems," *IEEE Security & Privacy*, vol. 15, no. 5, pp. 60–68, 2017.
- [26] T. Good and M. Benaissa, "AES on FPGA from the fastest to the smallest," in *CHES, Springer*, pp. 427–440, 2005.
- [27] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," in Proc. of the 28th Annual ACM Symposium on Theory of Computing, pp. 212–219, 1996.
- [28] W. Ali and J. Qadir, "A compact AES encryption core on Xilinx FPGA," in *2009 2nd International Conference on Computer, Control and Communication*, Karachi, Pakistan, pp. 1-4, 2009.
- [29] U. Khalid, M. A. Imran, and M. I. Abbas, "A compact and efficient AES-32GF for encryption in small IoT devices," *Multimedia Tools and Applications*, vol. 83, pp. 10609–10626, 2024.
- [30] X. Zhang et al., "VLSI Architecture of S-Box With High Area Efficiency Based on Composite Field Arithmetic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1597-1601, 2022.

ABOUT THE AUTHOR



Tran Van Khanh

Workplace: Institute of Cryptographic Science and Technology, Vietnam Government Information Security Commission
Email: tvkhanh@bcy.gov.vn

Education: He received the B.Sc. degree in 2009, the M.Sc. degree in 2011, and the Ph.D. degree in 2015 from the Moscow Institute of Physics and Technology (MIPT), Russian Federation.

Recent research direction: His research interests include electronic engineering, cryptographic engineering, hardware security, and embedded cryptographic systems.

Tên tác giả: **Trần Văn Khánh**

Cơ quan công tác: Viện Khoa học - Công nghệ mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: tvkhanh@bcy.gov.vn

Quá trình đào tạo: Tốt nghiệp Cử nhân năm 2009, Thạc sĩ năm 2011 và Tiến sĩ năm 2015 tại trường Đại học Vật lý kỹ thuật Mátxcova (MIPT), Liên Bang Nga.

Hướng nghiên cứu hiện nay: Kỹ thuật điện tử, kỹ thuật mật mã, bảo mật phần cứng và hệ thống mật mã nhúng.



Phan Van Ky

Workplace: Institute of Cryptographic Science and Technology, Vietnam Government Information Security Commission

Email: pvk.hvktqs@gmail.com

Education: He received his B.S. degree from Saint Petersburg Electrotechnical University "LETI" in 2013 and his M.S. degree in 2017. Recent research direction: His research interests include SoC design, hardware design, and hardware security.

Tên tác giả: **Phan Văn Kỳ**

Cơ quan công tác: Viện Khoa học - Công nghệ mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: pvk.hvktqs@gmail.com

Quá trình đào tạo: Tốt nghiệp Cử nhân năm 2013 và Thạc sĩ năm 2017 tại Đại học Kỹ thuật Điện Saint Petersburg "LETI".

Hướng nghiên cứu hiện nay: Thiết kế SoC, thiết kế phần cứng và bảo mật phần cứng.



Vu Van Viet

Workplace: Institute of Cryptographic Science and Technology, Vietnam Government Information Security Commission

Email: vietvu1912.97@gmail.com

Education: He received his B.S. degree from Le Quy Don Technical University in 2020 and is currently pursuing an M.S. degree at the Academy of Cryptography Techniques.

Recent research direction: His research interests include electronic engineering, hardware security, and hardware design.

Tên tác giả: **Vũ Văn Việt**

Cơ quan công tác: Viện Khoa học - Công nghệ mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: vietvu1912.97@gmail.com

Quá trình đào tạo: Tốt nghiệp Kỹ sư Đại học Kỹ thuật Lê Quý Đôn năm 2020 và hiện đang học Thạc sĩ tại Học viện Kỹ thuật mật mã.

Hướng nghiên cứu hiện nay: Kỹ thuật điện tử, bảo mật phần cứng và thiết kế phần cứng.