AI-Guided Lightweight Selective Encryption Framework for NAL-Based Video Streams in Real-Time Systems

YongkKtun Kim
Cyber Security Research Division
ETRI
Daejeon, South Korea
ykkim1@etri.re.kr

Geon Woo Kim
Cyver Security Research Division
ETRI
Daejeon, South Korea
gwkim@etri.re.kr

***Abstract*—This paper presents a theoretical framework for selective encryption of video streams, integrating AI-guided detection, video codec layer understanding, and lightweight cryptography. By analyzing the structure of H.264/AVC and H.265/HEVC streams and applying selective encryption at the Network Abstraction Layer (NAL), the approach aims to achieve content-aware privacy protection without compromising real-time performance. The theoretical contributions include a model for mapping object detection results to codec units and a formal outline for applying lightweight ciphers like ChaCha20 and HIGHT in embedded systems.**

Keywords—selective encryption, lightweight cryptography, H.264, HEVC, NAL, privacy, embedded systems

I. INTRODUCTION

As the use of smart video systems grows across domains such as surveillance, autonomous vehicles, and remote healthcare, ensuring privacy in video streams becomes critical. Full encryption of video data using traditional algorithms (e.g., AES) is computationally intensive and often impractical for resource-constrained edge devices. Furthermore, indiscriminate encryption of entire streams may disrupt standard decoding pipelines and lead to excessive energy consumption.

Selective encryption addresses these challenges by focusing cryptographic efforts on the most privacy-sensitive portions of the stream. However, traditional selective encryption techniques rely on static heuristics or predefined regions, which may not adapt well to dynamic environments. To enhance this paradigm, we explore a theoretical model that leverages object detection using deep learning to dynamically guide encryption policies.

This paper proposes a conceptual design that couples real-time object detection (e.g., YOLOv5-tiny) with codec syntax analysis (e.g., NAL unit parsing in H.264/HEVC) to perform lightweight, targeted encryption using algorithms like ChaCha20 or HIGHT. We present a layered view of how visual sensitivity maps can be transformed into encryption commands at the bitstream level.

II. BACKGROUND AND THEORETICAL FOUNDATIONS

A. Video Codec Structure and NAL Units

Modern video codecs such as H.264/AVC and H.265/HEVC utilize a modular bitstream structure composed of Network Abstraction Layer (NAL) units. These units encapsulate slices of macroblocks or coding tree units, along with auxiliary information like headers and parameter sets. Because visual content is tightly coupled to these units, selectively encrypting parts of NALs can obscure meaningful data while maintaining decoder compliance.

In our framework, encryption is applied to video coding layer (VCL) NAL units associated with identified regions of interest. We define a theoretical mapping function $f : R \rightarrow N$, where R denotes bounding boxes from detection results, and N represents NAL unit byte ranges. The goal is to approximate f such that encryption of $f(R)$ maximizes perceptual obfuscation while minimizing bitrate and latency cost.

B. Object Detection as Driver of privacy Policy

The use of AI models like YOLOv5 enables real-time object detection on embedded platforms. Each detected object is associated with a bounding box $Bi = (x_i, y_i, w_i, h_i)$, which corresponds to a spatial region in the frame. These coordinates must be translated into macroblock indices and eventually to NAL byte offsets. We define a region-to-slice projection $P : B_i \rightarrow S_j$, where S_j is a slice covering the spatial region.

This mapping requires calibration between pixel resolution, frame buffer layout, and codec-specific macroblock tiling. The theoretical model assumes a known tiling scheme and deterministic motion estimation to simplify mapping. Given P and f , encryption commands can be dynamically generated for each frame.

C. Lightweight Cryptography for Real-Time Video

Stream ciphers such as ChaCha20 are particularly suitable for real-time applications due to their low computational overhead and resistance to timing attacks. Block ciphers like HIGHT are optimized for low-power platforms and support various block modes (CTR, CBC). To ensure format compliance, encryption is restricted to payloads of VCL slices, avoiding start codes and headers.

We define the encryption transformation as: $Ek\ (d) = c\ for\ d \in f(R)$where k is a frame-specific key d, is raw byte data from the stream, and c is the encrypted ciphertext. Security analysis assumes ephemeral keying and periodic IV

rotation.

III. Experiment and Consideration

The proposed system consists of five key modules operating in a real-time processing pipeline:

A. System Overview

We propose a layered theoretical architecture:

Layer 1: Visual Context Inference — Deep learning model infers sensitive content R.

Layer 2: Codec-Aware Mapping — Map R to byte stream targets N via P and f.

Layer 3: Policy Engine — Determine encryption mode (e.g., skip, full, partial) based on confidence scores or category priority.

Layer 4: Encryption Kernel — Apply lightweight ciphers to selected byte ranges.

Layer 5: Stream Reconstruction — Merge encrypted segments back into bitstream, ensuring decoder compatibility.

This framework allows future extensions such as:

Incorporation of scalable video coding (SVC) to adjust encrypted detail by bandwidth. Object-priority encryption (e.g., faces > plates > persons). Integration with key exchange protocols for secure session management. The platform is an RK3588 Android board with NNAPI acceleration.

B. Performance Evaluation

TABLE I. EXPERIMENTAL RESULTS

Method	Latency (ms)	Power (W)	PSNR (dB)	Entropy (b/B)	FPS
Full AES	43.2	3.4	36.1	6.21	19.2
Selective AES	16.7	2.0	36.0	7.83	28.9
ChaCha20	11.5	1.8	35.9	7.9	30.1
HIGHT	13.3	1.7	35.8	7.77	29.7

The experimental evaluation was conducted on an RK3588 Android board with NNAPI acceleration enabled. The platform was chosen to represent a practical embedded environment with constrained power and processing resources. The results in Table I demonstrate that ChaCha20 achieved the best latency and frame rate due to its stream cipher design, which minimizes block-level operations and leverages efficient bitwise computations. In contrast, HIGHT consumed the least power because of its optimized round function for low-resource devices. These findings validate that lightweight ciphers can achieve both high performance and energy efficiency when integrated with AI-guided selective encryption, making the proposed framework practical for real-time embedded video systems.

IV. Conclusion

This work introduces an efficient AI-driven selective encryption system tailored for real-time video on embedded devices. It demonstrates how combining object detection with codec-aware encryption yields strong privacy protection at low cost. Future work includes adversarial robustness and adaptive prioritization by object class.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00394190, Development of an open video security platform technology with on-device self-security by design).

References

[1] [1] L. Tang, "Methods for encrypting and decrypting MPEG video data
[2] efficiently," U.S. Patent 6,792,080, 2004.
[3] [2] M. Qiao et al., "Selective encryption for H.264 video coding based on
[4] syntax elements," IEEE Trans. Consumer Electron., vol. 61, no. 4, pp.
[5] 551–558, 2015.
[6] [3] S. Kim et al., "CU-based ROI encryption in HEVC," in Proc. ICME,
[7] 2020. [4] J. He et al., "ROI-based perceptual video encryption," IEEE
[8] Access, 2020.
[9] [4] G. Jocher et al., "YOLOv5," 2021. [Online]. Available:
[10] https://github.com/ultralytics/yolov5
[11] [5] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement,"
[12] 2018.
[13] [6] D. J. Bernstein, "ChaCha, a variant of Salsa20," 2008.
[14] [7] S. Hong et al., "HIGHT: A new block cipher suitable for low-resource
[15] device," in Proc. CHES, 2006.
[16] [8] K. Lee et al., "Secure and efficient HIGHT implementation on ARM,"
Electronics, 2018.