SmartCamID: Real-Time Camera Stream Encoding for Live Broadcasting

Aditi Atul Karad
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623
ak2298@rit.edu

***Abstract*—Live streaming has evolved to be a pervasive medium for processing up-to-the-minute content in various domains, including sports, education, and entertainment. SmartCamID aims to address arguably the pressing challenges in live broadcasting through the provision of a system that offers secure embedding of metadata, multi-camera feed handling, and low-latency streaming. By using applications like OBS Studio, Fernet encryption, and a scalable and secure Unity Proxy, it is introduced to support continuity with the project method of processing real-time video. This paper explores the methodology, technical novelties, and experimental results that highlight the system's efficiency and adaptability in modern broadcasting workflows.**

I. Introduction

Modern communication depends on live broadcasting with an array of applications from sports to education to corporate sports and entertainment. As audiences demand higher streams and interactive features, the broadcast industry faces challenges posed by proprietary video content, maintaining low latency, and accurately identifying video sources in real time. Virtual camera technology and metadata embedding offer creative opportunities to improve broadcast workflows. So, this project SmartCamID aims to bring these advancements into a robust, scalable system for streaming — encoding a stream of real-time video. By integrating encrypted metadata into video streams, broadcasters can ensure source identification, and secure and optimize workflows, making the system perfect for platforms like YouTube, Zoom, and Twitch.

Key objectives include:

- Incorporating smooth integration with hardware like iPhones and laptops for capturing video feed.
- Metadata can be embedded securely without affecting video quality or adding significant latency.
- Enabling compatibility with open-source streaming tools like OBS Studio to ensure adaptability across platforms.

II. Background

The advancement of live broadcasting relies on developments in hardware and software technologies. Traditional broadcasting systems tend to utilize proprietary hardware, which tends to be costly and rigid. The rise of software-based options, like OBS Studio and virtual cameras, has made high-quality streaming tools available to all, making them accessible for a wider audience. But there are still challenges in real-time security and metadata management.

Metadata plays a critical role in identifying video sources, ensuring accountability, and allowing for streamlined content management. Embedding metadata securely and seamlessly into live video streams while maintaining performance is a major technical challenge. On top of that, macOS's Continuity Camera added a new way of using devices like iPhones as high-quality webcams. While beneficial, directly merging this functionality in real-time encryption and streaming workflows is challenging in terms of compatibility and performance

The project aims to solve this by making a universal, secure, and effective live broadcasting solution.

III. Technical Overview

This section introduces the tools and technologies used in the project.

A. OBS Studio

OBS Studio is free, open-source software for video recording and live streaming. It supports high-performance live video and audio capture and mixing. In this project, OBS Studio is used as the last step in the transmission pipeline, making it compatible with platforms like YouTube, Zoom, and Twitch via its virtual camera and custom streaming configurations.

B. Fernet Encryption

Fernet is a symmetric encryption method provided by the Python Cryptography library. It ensures the confidentiality of metadata by encrypting it before embedding it into video streams. This encryption method is ideal for maintaining security without adding significant computational overhead.

C. Metadata in Video Streams

Metadata is the extra contextual information coded into video streams relevant for each query, including the source identifier, timestamp, and session ID. All videos come with some metadata, like resolution, rate, and codec information, but in our project, custom metadata is added to identify sources and enhance security.

D. Continuity Camera

With Apple's Continuity Camera feature, iPhones can function as high-quality webcams for macOS devices. It uses Wi-Fi and Bluetooth for smooth connection and superior video quality compared to most built-in laptop cameras. This project uses Continuity Camera for feed capture due to its ease of use and high resolution.

IV. Methodology

The SmartCamID methodology is designed to deliver a robust system for real-time video processing and streaming. It is divided into three primary components:

A. Feed Capture

The feed capture module integrates multiple video sources from the feed capture module, including:

- iPhones linked through macOS's Continuity Camera feature for very high-resolution video feeds.
- Integrated laptop cameras and USB external webcams.
- Dynamic camera switching during live broadcasts to ensure flexibility.

OpenCV is utilized to process the feeds, allowing near real-time video recording and preprocessing for upcoming stages.

B. Metadata Encryption

The metadata encrypted module ensures every video stream has a unique, secure identifier:

- Encryption via the Fernet library protects the metadata so it is private and cannot be manipulated.
- Metadata, including the camera source, date and timestamp, and session ID, is directly embedded into video frames.
- The embedding process is designed to keep overhead minimal while remaining real-time.

C. Streaming via OBS Studio

Processed video streams are passed to OBS Studio, a widely-used broadcasting tool:

- Integrations with virtual cameras allow processed feeds to be recognized as standard video sources.
- Configurations are fine-tuned to ensure compatibility with popular platforms like YouTube, Zoom, and Twitch.
- OBS Studio handles flexible streaming configurations, like setting message durations and encoding video formats.

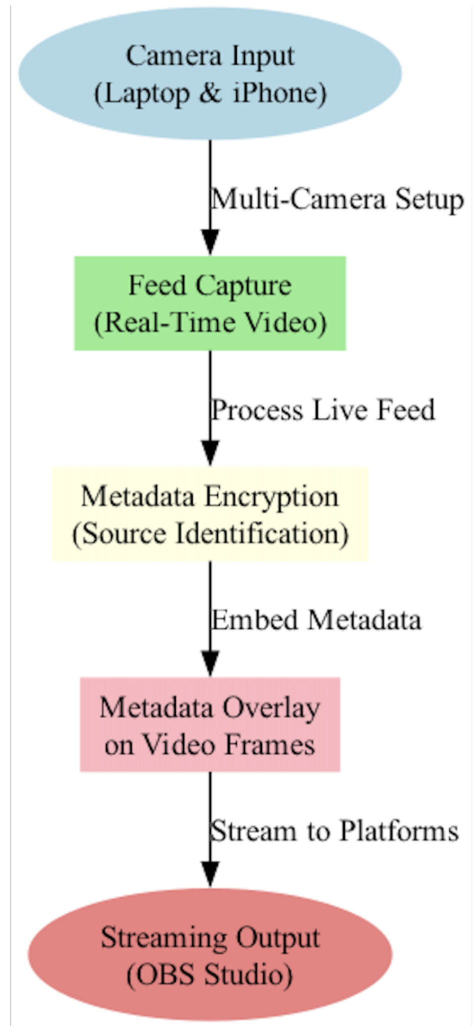

Fig. 1. Overall Workflow of the SmartCamID System

V. Results

A. Performance Metrics

The system's performance was evaluated on key metrics:

- Average metadata embedding time: 3ms per frame.
- Latency introduced by the entire pipeline: 50ms, ensuring a smooth viewer experience.
- Compatibility with resolutions up to 4K, maintaining high-quality video output.

B. Platform Testing

The system was tested across multiple platforms, with the following results:

TABLE I
PLATFORM TESTING RESULTS

Platform	**Latency Observed (ms)**	**Resolution Quality**
YouTube	200	High
Zoom	100	Moderate
Twitch	150	High

a) Explanation of the Table:: The table represents the latency and resolution quality on three platforms: YouTube, Zoom, and Twitch. YouTube provides the best resolution quality but with higher latency (200ms), which is tolerable for live streaming where quality matters. Zoom achieves the lowest latency (100ms), making it ideal for real-time interactions like video conferencing but with lower resolution quality. Twitch balances resolution quality and latency (150ms), making it suitable for various streaming purposes. These results show how different platforms handle latency and resolution quality based on use cases and provide insights for optimizing configurations for different platform requirements.

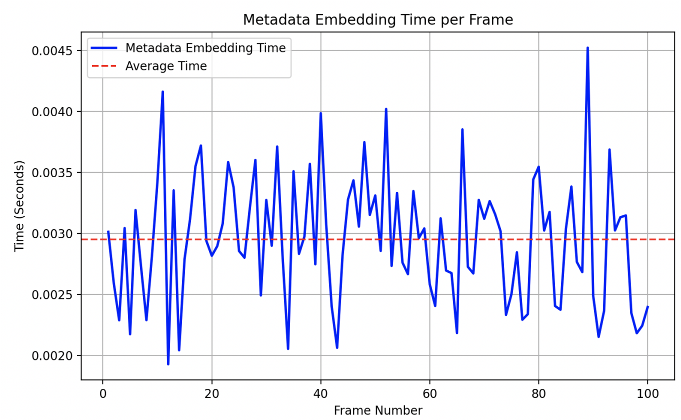

Fig. 2. Metadata Embedding Time per Frame shows consistent and low processing times, averaging 3ms, ensuring real-time performance

Fig. 3. Metadata Embedding Time per Frame shows consistent and low processing times, averaging 3ms, ensuring real-time performance

b) Graph "Metadata embedding time per frame" shows how long it takes to write metadata on a single video frame in real time. The blue line represents the embedding time across frames, and the red dashed line shows the average time over all frames. The graph demonstrates stable performance, with most frames embedding times around 3 milliseconds, on average. Rare peaks exceed 4ms, possibly due to system interruptions or extra computational loads. However, these differences minimally affect embedding times, remaining acceptable for real-time applications, ensuring smooth integration into live video streaming workflows. This performance showcases the effective and reliable process of embedding metadata in low-latency operations.:

c) High-Latency vs. Resolution Trade-Off Graph illustrates the relationship of Latency and Quality of Resolution for two popular platforms: YouTube and Zoom. YouTube offers top-definition streaming, normalized to a mediation quality value of 1.0; however, this comes at the cost of increased latency (200ms), making it not ideal for real-time interactions. Conversely, Zoom requires low latency (100ms), essential for interactive applications like video conferencing, but at the trade-off of normalized resolution quality of 0.7. The graph shows platform adjustments based on their specific use-case priorities — YouTube focuses on content quality, while Zoom prioritizes real-time responsiveness. This trade-off aids in guiding configuration tuning to align with streaming needs.:

VI. Challenges

Several challenges were encountered and addressed during the project:

- Bypassing macOS security restrictions to activate the OBS virtual camera feature.
- Adding security through encryption and embedding metadata without interrupting real-time workflows.
- Resolving video compatibility issues with resolutions and frame rates.

TABLE II
Process Steps and Challenges

Process Step	**Tools/Technologies Used**	**Challenges**
Feed Capture	OpenCV, Continuity Camera	macOS Continuity Camera setup
Encryption	Fernet (Python)	Real-time performance optimization
Streaming	OBS, pyvirtualcam	macOS restrictions, OBS latency

a) Explanation of the Table II:: The table lists the essential elements of the SmartCamID system, emphasizing the tools used and hurdles faced in each step. Feed capture is powered by OpenCV and Continuity Camera for combining multiple video sources, but macOS-specific configuration issues posed initial hurdles. Encryption is performed with the Fernet library for efficient embedding of secure metadata, requiring real-time performance optimization. Lastly, OBS and pyvirtualcam facilitate streaming to multiple platforms, though configuration challenges exist due to macOS restrictions. These actions showcase a comprehensive approach to solving real-time broadcasting challenges.

VII. Conclusion

A. Achievements

Real-time integration of the SmartCamID system: still images, video capture, metadata encryption, and streaming into a seamless workflow. It has shown its capacity to accomplish the primary goals of secure, effective, and flexible live broadcasting, which makes it a significant contribution to modern streaming technologies. Key achievements include:

- **Secure Metadata Embedding:** Implemented a tangible technique for securely embedding metadata into video streams without sacrificing the quality of processing or adding substantial processing delays.
- **Multi-Camera Support:** Implemented and controlled video feeds from several cameras, including top-notch Continuity Camera feeds and laptop cameras, offering additional versatility for diverse broadcasting setups.
- **Low Latency Performance:** Achieved minimal latency during live broadcasts, ensuring a smooth and uninterrupted viewer experience, critical for applications such as live sports and interactive sessions.
- **Cross-Platform Scalability:** The system was validated on prominent platforms such as YouTube, Zoom, and Twitch, signaling its versatility touches a plethora of real-time environments and use cases.
- **Ease of Integration:** Leveraged widely-used tools such as OBS Studio and OpenCV, enabling easy adoption and customization for future use cases.

B. Future Scope

The initiative lays grounds for additional innovation and optimization with scaling needs provision: The demand for live broadcasting in various domains. Future directions include:

- **Enhanced Scalability:** Growing the run-time for large-scale broadcasting environments like multi-location live events by optimizing metadata for higher volumes handling and feed processing.
- **Hardware Acceleration:** Leveraging GPU architecture, the new hardware acceleration, and virtual hardware encoding techniques enhance the efficiency of the entire system, especially at high-resolution video feeds.
- **AI Integration:** Adopting AI-based analytical tools to facilitate content moderation and improve metadata generation, including real-time audits such as engagement metrics and user behavior anomaly detection.
- **Native Platform Integration:** Integration of native macOS, Windows, and other applications or plugins for Linux, providing tighter integration with system-level features to enhance the broadcasting workflow.
- **Dynamic Resolution and Bitrate Adjustment:** Adaptive streaming monitoring improves video quality and bitrate according to network conditions, ensuring a stable performance across varying bandwidths.
- **Security Enhancements:** Improving encryption algorithms and using multi-factor authentication for secure live broadcasts that prevent unauthorized use.
- **Interactive Features:** Built for chat, live polls, social polls, and synchronized audience feedback to enhance user engagement during broadcasts.

These developments will cement SmartCamID as an all-in-one solution for security, efficiency, and dynamic live broadcasts in an ever-changing digital landscape.

VIII. References

References

[1] OpenCV Documentation. [Online]. Available: https://docs.opencv.org

[2] Python Cryptography Library. [Online]. Available: https://cryptography.io/en/latest/

[3] OBS Studio User Manual. [Online]. Available: https://obsproject.com/wiki/

[4] AVFoundation Programming Guide. [Online]. Available: https://developer.apple.com/documentation/avfoundation/