Enhancing Drone Video Analytics Security Management using an AERPAW Testbed

Alicia Esquivel Morel *, Zack Murry *, Kevin Kostage †, Chengyi Qu ‡, Prasad Calyam*
*Department of Electrical Engineering and Computer Science, University of Missouri - Columbia, USA.
† Department of Computing and Software Engineering, Florida Gulf Coast University - Florida, USA.
Email: *{ace6qv, zjmfrr, calyamp}@missouri.edu, †kskostage9457@eagle.fgcu.edu, ‡cqu@fgcu.edu

***Abstract*—Drone-based applications involving real-time video analytics are emerging to serve diverse situational awareness use cases ranging from precision agriculture to disaster response. Hence, there is a need to study robust security measures to ensure the integrity of information and safeguard communication, as well as data transmission in drone video analytics. In this paper, we investigate methods to enhance the security management of drone video analytics in terms of reliability and integrity within realistic settings using testbed resources in the NSF-supported AERPAW infrastructure. Specifically, we study security mechanisms to model and detect threats such as Replay, Packet Injection, and Physical Capture attacks caused by situations in dynamic and potentially adversarial network environments. In addition, we generate balanced datasets through Generative Adversarial Networks (GAN) to address challenges posed by unbalanced datasets that are common when applying machine learning models for attack detection impacting drone video analytics traffic. Our experimental environment in AERPAW involves a setup for secure communication through a MAVLink-based (open-standard) drone communication protocol that uses continuous authentication via digital signatures. Our experiment results compare the efficiency gains achieved through secure MAVLink-based communication with unsecured counterparts, examining factors such as packet encryption, digital signatures, and nonces. Further, our results provide valuable insights into the adaptability of security mechanisms for drone video analytics within realistic environments.**

***Index Terms*—Drone-based applications, security management, attack detection, threat impact mitigation, experimental testbed**

I. Introduction

Drone-based applications have been increasingly employed in various use cases, encompassing both civilian and military applications, spanning from precision agriculture to communication coverage extension [1]. In addition, drone systems can be embedded with high-resolution video cameras and deployed in real-time scenarios, where drones can be used for situational awareness tasks such as video surveillance, object detection, or tracking [2]. Simultaneously, video analytics with drones can play a crucial role in applications such as search and rescue, traffic management in smart cities, and disaster response [3].

In addition, these systems may face cyber threats, including GPS spoofing attacks, Distributed Denial-of-Service (DDoS) attacks, and Man-In-The-Middle (MITM) attacks, which could potentially compromise the integrity of collected information, due to these systems' unique network and dispersed physical systems that can be located in remote places [4]. Thus, there is a need to study robust security measures that safeguard communication and data transmission within these real-time applications and ensure the success of mission tasks by protecting against unprecedented accidents or potential attacks impacting drone video analytics.

In this paper, we investigate methods to enhance the security management of drone video analytics in terms of reliability and integrity within realistic settings using testbed resources in the NSF-supported Aerial Experimentation and Research Platform for Advanced Wireless (AERPAW) infrastructure [5]. The AERPAW, as the pioneering wireless research platform, is dedicated to exploring the convergence of 5G technology and autonomous drones, showcasing its relevance and benefits to our proposed research. Specifically, we study security mechanisms to model, detect and defend against diverse and sophisticated threats such as Replay, Packet Injection, and Physical Capture attacks caused by situations in dynamic and potentially adversarial network environments using a STRIDE-Per-Element [6] threat model considering each of the drone video transmission elements. In addition, we generate balanced datasets through Generative Adversarial Networks (GAN) [7] to address challenges posed by unbalanced datasets that are common when applying machine learning models for accurate attack detection impacting drone video analytics traffic.

For our study, we set up an experiment in AERPAW that involves secure communication through a Micro Aerial Vehicle Link (MAVLink)-based (open-standard) drone communication protocol [8] over wireless channels using continuous authentication via digital signatures. Notably, MAVLink is susceptible to various attacks due to the absence of inherent security measures, and the protocol lacks native support for confidentiality and authentication mechanisms [9]. To address these limitations, our study considers proactive measures to secure data communication with MAVLink packets and enables the protocol to establish continuous authentication, thereby enhancing the overall security of drone video analytics.

In our AERPAW experiment, we emulate real-world conditions while designing a controlled environment for our experimentation of security management for drone video analytics. Moreover, we leverage advanced digital signatures to ensure the continuous verification of the integrity of communication

This material is based upon work supported by the National Science Foundation under Award Numbers: CNS-1950873, CNS-1647182 and OAC-2018074. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

between drones, or between drones and the Ground Control Station (GCS). Our experiment results from the AERPAW testbed compare the efficiency gains achieved through secure MAVLink-based communication with unsecured counterparts, examining factors such as packet encryption, digital signatures, and nonces. Further, our results provide valuable insights into the adaptability of security mechanisms for drone video analytics within realistic environments.

The remainder of the paper is organized as follows: Section II provides the background and related work. In Section III, we describe the deployment and network environment setup for the experimentation with a video streaming application. Section IV details our security management study. Section V describes the testbed experiment results and salient findings. Lastly, Section VI concludes the paper.

II. Background and Related Work

In this section, we first introduce and motivate drone video analytic applications. Next, we introduce AERPAW benefits for our security management study.

A. Security in Drone Video Analytics Applications

Drone video analytics has revolutionized visual situational awareness use-cases in precision agriculture e.g., to support efficient farming practices, providing insights into crop health, disease detection, or yield prediction, informing farmers' decision-making [10]. It also has helped bolster disaster response efforts e.g., to improve search and rescue operations in scenes of interest where first responder mobility is limited [2]. Thus, drone video analytics plays a crucial role in achieving efficiency and foresight in critical mission tasks involving edge computing and wireless communications [11]. However, there is currently a limited understanding of how hackers can carry out cyber attacks to take control of drones, leading to interception or even crashes that impact mission success. Drones can also be exploited for malicious purposes through physical hijacking to corrupt software or datasets in the field. Therefore, it is crucial to study and develop novel methods to accurately detect and effectively prevent such attacks to mitigate potential damages [12]. The methods for securing drone video analytics need to ensure the integrity of information and safeguard communication and data transmission in critical mission tasks. Moreover, they need to feature advanced encryption protocols to enable secure communication channels.

B. AERPAW Benefits for our Experimental Research

Real-world experimentation with drones can present challenges due to their complex nature or regulatory and safety concerns imposed by aviation regulations. These challenges restrict how and where drones can be operated, consequently limiting the scope of experimentation. Furthermore, the integration of drones into complex applications requires seamless communication and coordination, in addition to a robust and secure network infrastructure, adding extra layers of complexity. To address these challenges, there is a need for a controlled experimentation platform that can facilitate a realistic environment, not only to tackle these complexities but also to investigate potential vulnerabilities in communication channels and ensure that data integrity is preserved. The AERPAW infrastructure offers unique capabilities to explore such security concerns, and supports conducting experiments in real-world scenarios, playing an important role in investigating security threats and leveraging insights gained through cutting-edge technology integration.

III. AERPAW Experiment Configuration

In this section, we first provide details on the deployment and operation environment setup in AERPAW. Subsequently, we describe our drone video analytics application used in our AERPAW experiments.

A. Operation Deployment Setup

In leveraging AERPAW for the enhancement of drone video analytics-related security management, our operational development setup adheres to the batch-mode access provided for all canonical (Program-it-Yourself) experiments within the AERPAW infrastructure. Experimenters are tasked with fully developing their experiments within the virtual environment, recognizing that AERPAW is primarily designed to operate in batch mode rather than as a live model. This operational approach necessitates the preparation and submission of experiments, which are subsequently executed at a later time when the required resources become available.

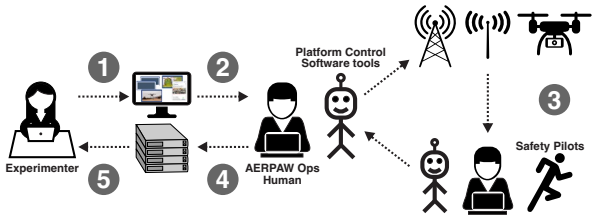


Figure 1: Five sequential steps guide the experimenter in AERPAW experiment development process that includes: *i)* Beginning in the Virtual Digital Twin Environment, the experimenter develops the experiment and submits testbed execution requests upon completion, *ii)* AERPAW Ops/Control then transitions the experiment to the physical testbed, *iii)* execution, led by AERPAW Ops/Control and safety pilots, follows on schedule, *iv)* post-physical experimentation, the experiment seamlessly returns to the virtual environment, and *v)* the experimenter reviews results in AERPAW's virtual digital twin, enabling iterative refinement and re-submission.

The deployment and network environment for our experiment involves a specific setup or development process that must be followed before submitting experiments for execution on the AERPAW physical testbed. **Figure 1** outlines AERPAW's usage workflow, encompassing both the virtual (digital twin) and physical (testbed) environments. The initial step requires users to design the experiment through the experimenter portal, creating an initial definition. During this phase, users must specify the subset of AERPAW fixed nodes to be included in the experiment, considering the unique geographical locations of these nodes. This structured workflow ensures a systematic and effective development process for our experiment within the AERPAW infrastructure.

In our operational setup for leveraging AERPAW to enhance drone video analytics security management, we specify the required portable nodes for the experiments' configuration. Once the experiment design is complete, we submit it for execution by AERPAW operations personnel (Ops), who schedule it for a time when the necessary resources are available. The results are then received through the virtual environment, allowing for a comprehensive evaluation of drone video analytics performance and security within the AERPAW infrastructure.

B. Experimentation Deployment Setup

The first step in the deployment of the environment setup is based on the FlyPaw work in [13]. We use AERPAW's software libraries to instruct a drone to follow a pre-planned trajectory from a `.plan` file. Second, once this runs, AERPAW's finite state machine functionality can be employed to fulfill custom requirements when the vehicle is in transit or stopped at a way-point by transitioning to a certain state to execute appropriate actions. Third, we start with a development environment, which is executed through the Operator Experiment Oversight (OEO) Console, connected to a local QGroundControl [14] via an SSH tunnel.

Our experimentation is based on nodes LPN1 (A large drone node) and LW1 (A fixed radio node at the Lake Wheeler site), as pictured in **Figure 2**. Our experiment deployment is distributed on a specific set of AERPAW Radio Nodes (ARN) in the testbed environment. The first step is the experimentation mode in the AERPAW Virtual Node (AVN) environment, which provides a virtual machine instance for each node with an additional instance for the OEO Console. Experimenters have live access to develop their experiments here. Security measures are implemented in this stage. Consequently, a custom-designed emulation of the radio environment and vehicle mobility is integrated into this environment. This allows the software developed by the experimenter to run seamlessly. As AERPAW is a batch-mode processing facility, after experiments are done in the AVN, the experiments are later run over the actual AERPAW fixed and portable radio nodes in the field. Lastly, our experimental results are recorded and transmitted back into the development virtual nodes for analysis. The communication within the nodes is accomplished with MAVLink as a lightweight and efficient protocol. This aims to connect the various components of the experiment and transmit telemetry data. In addition, this serves as command control. For the autopilot software, ArduPilot [15] is used as a software suite for managing vehicle control.

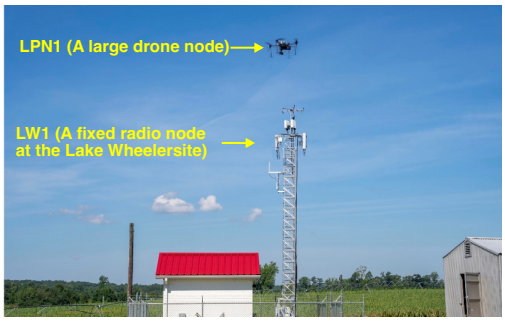


Figure 2: LPN1 and LW1 nodes used in experiments at Lake Wheeler site.

The small size file features a lower resolution, minimizing the overall data size, with 360p resolution, 12fps frame rate. Moderate video compression is applied, maintaining an acceptable quality. In contrast, for large video files, with 720p resolution, 15fps frames rate. We stressed the system's capabilities with higher resolution, frame rate, and duration of the file. Minimal compression techniques are applied to assess how well the system handles varying levels of security measurements. The video data is transmitted from the drone to the GCS, and the video data is processed before being displayed as a situation-specific video feed to the operator. Other additional information is also displayed in terms of e.g., status, GPS coordinates, and battery levels, which enhances the overall situational awareness for both the drone and the operator.

IV. Security Management in our AERPAW Experiment

In this section, we describe the attack vectors and perform threat modeling for the drone video transmission elements. Subsequently, we present algorithms for diverse attack detection and describe our approach to enhance security measures through use of balanced datasets obtained through GAN.

A. STRIDE Threat Modeling and Attack Detection

We performed an analysis involving a threat model based on Spoofing (S), Tampering (T), Repudiation (R), Information Disclosure (ID), Denial of Service (DoS), and Elevation of Privilege (EoP). Specifically, we used the (STRIDE)-Per-Element [6] threat model in order to understand the attack vectors and the possible adversary consequences. **Table I** describes in detail the threat categories, elements, descriptions, and attack vectors that relate to the drone video transmission elements. Herein, we describe the attack vectors:

Replay Attacks: This refers to security threats where an attacker intercepts and maliciously re-transmits or duplicates a previously captured communication. Severe consequences of this attack include the cause of misinformation, loss of control over the drone, or unauthorized access. Sequential nonces are introduced as a mechanism to inhibit replay attacks. Video transmission applications often maintain a stateful counter based on the number of frames transmitted, so this value can be combined with the number of packets transmitted for each individual frame to produce a unique and incremental nonce value. In order to validate sequential nonces, the receiver only accepts those frames with nonces that are strictly greater than all the preceding nonces.

Packet Injection Attacks: These attacks pose security threats where an unauthorized entity sends extraneous packets by positioning themselves between the drone and the GCS via a Man-in-the-Middle (MITM) attack. Malicious actors may insert fake video data to mislead users about the video transmission. For example, attackers could transmit frames

Table I: STRIDE-Per-Element Threat Model considering each of the drone video transmission elements.

	Element	Description	Attack Vectors
S	External Entity	Attacker mimics GCS identity	Packet Injection
T	Data Flow	Modification of video data in transit	Replay, Packet Injection
R	Process	Denying involvement in video manipulation	-
I	Data Store	Unauthorized access to stored video data	Physical Capture
D	Data Flow	Disruption or denial of video transmission	-
E	External Entity	Unauthorized access to privileged functions	-

that indicate an attack in one area to divert attention from another. Alternatively, attackers could send fake frames to cover footage of physical intrusions. **Algorithm 1** describes the initialization and procedure for a packet injection attack detection through the use of a digital signature, which allows for continuous authentication by repeatedly verifying the identity of the sender.

Algorithm 1 Packet Injection Attack Detection during MAVLink-based Video Transmission

MAVLink Communication, Transmission Time Packet Injection Attack Detection Result
Initialization: $L \leftarrow$ Digital Signature Length
$K \leftarrow$ Sender's Public Key
Result $\leftarrow$ No Packet Injection Attack Detected
Procedure:
for *Current Time in Transmission Time* **do**
  Packet $\leftarrow$ Get Packet(Drone, GCS)
  Digital Signature $\leftarrow$ Get First N Bytes(Packet, L)
  Plaintext $\leftarrow$ Get Last N Bytes(Packet, Length(Packet) $-L$)
  Message Hash $\leftarrow$ SHA256 Hash(Plaintext)
  Signature Verified $\leftarrow$ Verify Digital Signature(Message Hash, Digital Signature, K)
  if *Signature Verified* **then**
    Accept Packet(Packet)
  end
  else
    Drop Packet(Packet)
    Result $\leftarrow$ Result $\oplus$ Packet Injection Attack Detected
  end
end

Physical Capture Attacks: these may lead to unauthorized physical interference with a drone by an external actor. They involve the threat agent gaining physical control of the drone, either by capturing it in flight or on the ground. This attack can allow the threat agent to capture images or videos, gain access to this information, or intentionally damage or destroy the captured drone, leading to financial losses and potential harm to other actors. **Algorithm 2** describes the initialization and procedure for a physical capture attack detection. We take the cross-product of the distance vectors between the drone and the waypoints at either end of its path to ensure that it stays in line with the expected path. Additionally, we allow the drone an accepted range around the two endpoints of its path.

Algorithm 2 Detecting Physical Capture Attacks during Drone Operation

Pos, Start, End PhysicalCaptureDetected
Initialization: $d \leftarrow$ DistanceThreshold $x \leftarrow$ CrossProductThreshold
Procedure:
CrossProduct $\leftarrow$ $(End.\text{Lat} - Pos.\text{Lat}) \times (Start.\text{Lon} - Pos.\text{Lon}) -$ $(End.\text{Lon} - Pos.\text{Lon}) \times (Start.\text{Lat} - Pos.\text{Lat})$
if $|$*CrossProduct*$| \leq x$ *or* distance(Pos, Start) $\leq d$ or distance(Pos, End) $\leq d$ **then**
  return False
end
return True

B. Enhancing Security Measures through the use of GAN

Our primary objective for this contribution of the paper is to assess the effectiveness of our security measures in real-world scenarios within the AERPAW testbed. This involves not only validating the efficacy of our security measures but also gaining valuable insights into expediting the detection of unsecured communication. In real-world scenarios, cyber-attack datasets typically display an imbalance, with normal traffic significantly outweighing malicious or intrusive traffic [16]. Such disproportion presents challenges in training models to recognize and learn patterns of malicious activity, which are relatively scarce. Consequently, models tend to develop a bias towards identifying the majority class, i.e., normal traffic. This bias, while not hindering the overall detection of intrusions or malicious activities, adversely affects the accurate classification of specific attack types. As we integrate our security measures for drone video analytics into the AERPAW experiment environment, we encounter similar challenges related to unbalanced and limited datasets concerning network traffic, cyber-attacks, and drone attacks. These challenges pose difficulties in accurately determining unsecured communication situations.

By leveraging GANs, we showcase the effectiveness of GAN-generated balanced datasets in bolstering security measures in dynamic and challenging drone application environments. The utilization of GANs empowers us to address data imbalances, ensuring that our security models are well-equipped to handle the complexities inherent in real-world scenarios. GAN utilizes two deep learning sub-models: the generator and the discriminator. The generator aims to produce the most realistic dataset possible, while the discriminator is trained to distinguish between realistic and generated datasets. The sub-models are trained based on each other's outputs, improving with each generation. This method surpasses the simple data augmentation techniques that cause the need to use over-sampling balancing methods and thus provides balanced datasets as well as mitigates issues related to overfitting. Since the GAN model can be retrained with new datasets, the technique can be adapted to evolving attack vectors.

Table II illustrates the results of a supervised learning analysis conducted on a sample unbalanced dataset comprising 100,000 samples. The dataset encompasses diverse attack traffic, facilitating a comparative examination of the precision between real attack traffic and generated attack traffic at a 20:7:3 ratio between normal traffic, traffic with injected packets, and Replay attack traffic. We can observe from the table that it becomes evident that attack detection models exhibit higher precision scores when trained on the imbalanced dataset compared to the balanced one. Examining the training speed, both training and prediction times are markedly lower with imbalanced datasets, except for the Random Forest model. This can be attributed to models having less data to train on for each minority class. However, the balanced dataset demonstrates faster prediction times, which is often more relevant in real-time attack detection.

Table II: Accuracy (precision and recall) and Run Time (in seconds) comparison results on Imbalanced (ID) and Balanced Dataset (BD) under various models, i.e., Random Forest (RF), Gradient Boosting Machine (GBM), Support Vector Classifier (SVC), K-Nearest Neighbor (KNN), and Naive Bayes (NB).

Model	Run Times		Normal Traffic		Packet Injection Traffic		Replay Attack Traffic	
	Training Time	Prediction Time	Precision	Recall	Precision	Recall	Precision	Recall
	I D / B D	I D / B D	I B / B D	I D / B D	I D / B D	I D / B D	I D / B D	I D / B D
RF	313 s / 269 s	0.46 s / 0.32 s	76.0 % / 49.2 %	98.8 % / 92.5 %	95.5 % / 87.8 %	48.5 % / 51.9 %	94.6 % / 95.7 %	94.5 % / 97.4 %
GBM	1.73 s / 252. 1 s	0.02 s / 0.31 s	76.0 % / 49.3 %	98.9 % / 92.9 %	95.6 % / 88.5 %	48.4 % / 51.7 %	94.3 % / 95.7 %	95.7 % / 97.5 %
SVC	262.7 s / 296.3 s	48.8 s / 0.38 s	75.2 % / 49.2 %	99.4 % / 92.7 %	98.4 % / 88.6 %	44.2 % / 51.9 %	89.1 % / 95.7 %	94.7 % / 97.5 %
KNN	0.030 s / 0.027 s	2.2 s / 2.0 s	75.0 % / 47.9 %	91.2 % / 49.7 %	74.5 % / 63.2%	46.3 % / 57.5 %	85.4 % / 89.3 %	95.5 % / 98.0 %
NB	0.076 s / 0.075 s	0.001 s / 0.0001 s	76.5 % / 49.3 %	92.2 % / 91.2 %	72.9 % / 83.6 %	52.9 % / 53.5 %	89.1 % / 96.1 %	96.0 % / 96.1 %

The utility of GANs extends to the detection of sophisticated cyber threats as well. They achieve this by generating diverse attack variations, which enhances the sensitivity of the models to these threats. In conclusion, the integration of GAN models enhances the efficacy of security measures used for drone video analytics in the AERPAW testbed by addressing challenges posed by unbalanced datasets and facilitating the accurate detection of diverse threats.

V. Testbed Experiment Results

As security measures are rarely used in isolation, our experiments are run in the context of the FlyPaw route-planning algorithm for scenarios with limited connectivity. The FlyPaw algorithm [13] minimizes the age of information by selectively backtracking to places with strong connections using a decision tree when it loses connection, as detected by `iPerf3` calls. Our experiments consist of a large drone node that transmits video data at three waypoints back to a fixed "base station" node (LW1). As two of these waypoints are outside of the base station's range of connectivity, the FlyPaw algorithm backtracks the drone to deliver the information. We conducted experiments to measure the effectiveness of responses to several attack scenarios in the AERPAW testbed.

A. Attack Scenarios

Replay: Both an unsecured system and a system secured using nonce values were tested against simulated Replay attacks. **Figure 3(a)** illustrates the number of packets received (either authentic or duplicated) during an attack that replays every packet in a vulnerable system in comparison to a secured one. In addition to the mitigation of these attacks using nonces, we show the corresponding packets received over time.

Packet Injection: We tested the efficacy of using Digital Signature Standard (DSS) fips-186-3 in the AERPAW testbed via simulating packet injection attacks against the video analytics setup. **Figure 3(b)** describes the number of frames received over time. During the attack, three fake frames were injected per real frame, including a null signature in the DSS setup.

Physical Capture: We simulated physical capture attacks on the drone by directly modifying AERPAW's software library to cause unexpected behavior. We tested how the interval at which **Algorithm 2** is run affects the delay in recognizing physical capture attacks, as shown in **Figure 3(c)**. While the use of smaller polling intervals detects attacks quicker, it can strain the drone's ability to perform other tasks.

B. Digital Signature and Encryption Algorithms in AERPAW

Packet Encryption Impact Analysis: Encrypting information is a necessary step in modern systems to maintain data privacy. We compare the encryption algorithms `ChaCha20`, Advanced Encryption Standard (AES), and RSA with PKCS#1 OAEP to assess their utility in drone-based video analytics, as shown in **Figure 4(a)** [17]. While `ChaCha20` is commonly chosen as an efficient algorithm, both AES and PKCS#1 yielded comparable results in our use case, as they all have minimal effects on both the mission time and the transmission rate of frames. PKCS#1 was found to cause a slightly larger reduction in the transmission rate, which is likely due to its transmission of 288 extra bytes per frame, but this was not enough to propagate into a larger mission time.

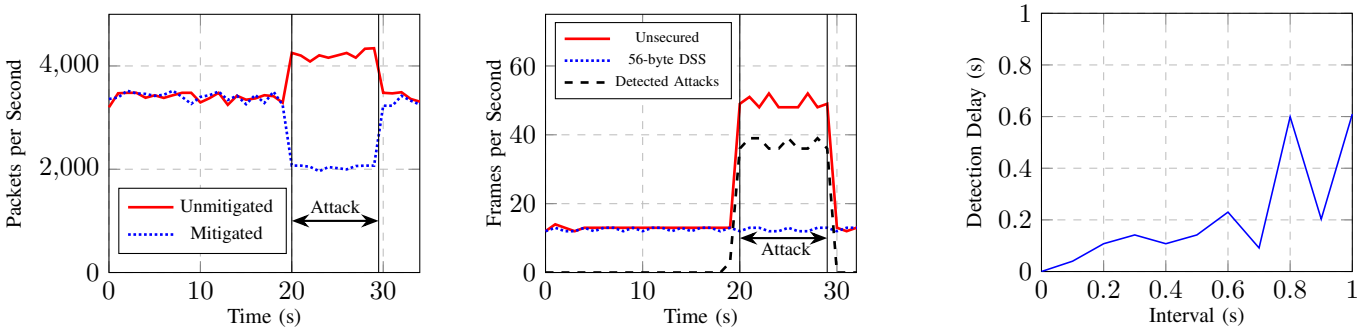
Digital Signature Assessment: To assess the computational overhead incurred through the use of continuous digital signatures, through our proof of concept implementation in AERPAW, we compare the following digital signature algorithms to the baseline unsigned performance: DSS fips-186-3, Elliptic Curve Digital Signature Algorithm (ECDSA), and PKCS#1 with 56-, 96-, and 256-byte digital signatures, respectively [18]. As shown in **Figure 4(b)**, the use of digital signatures was found to result in a substantial decrease in the speed of video transmission, but this effect was still minor in the wider context of the entire mission.

C. Analysis and Discussion

Our experiments suggest that the use of fundamental security measures (i.e., encryption, digital signatures, and nonces) in AERPAW does not significantly hinder drone-based mission objectives in video analytics scenarios. For example, while ECDSA was found to decrease the rate of video frame transmission by 19.06%, this figure only results in a modest increase in mission time due to the fact that a drone spends more time in flight rather than actively transmitting video data. This increase was found to be 0.93% and 2.26% for the scenarios with small and large videos, respectively. This difference can be attributed to the proportion of time being used to transmit video in the two scenarios involving small and large videos. Since the larger video requires more time to transmit, small decreases in the transmission rate result in a much greater impact on the mission time.

VI. Conclusion

In this paper, we addressed security needs in drone-based applications, specifically focusing on video analytics using the AERPAW testbed. We investigated security management to enhance video streaming reliability and integrity during critical missions. In addition, we implemented the MAVLink-based drone communication protocol for secure communication with



(a) Replay Attack

(b) Packet Injection Attack

(c) Detection Delay of Physical Capture

Figure 3: Attack scenarios for Replay, Packet Injection, and Physical Capture attacks. (a) Over the interval between 20 and 30 seconds, each package sent was duplicated to disrupt the system, (b) The number of frames received over time, in which three fake frames were injected per real frame, and (c) Delay in the detection of physical capture attacks based on polling interval.

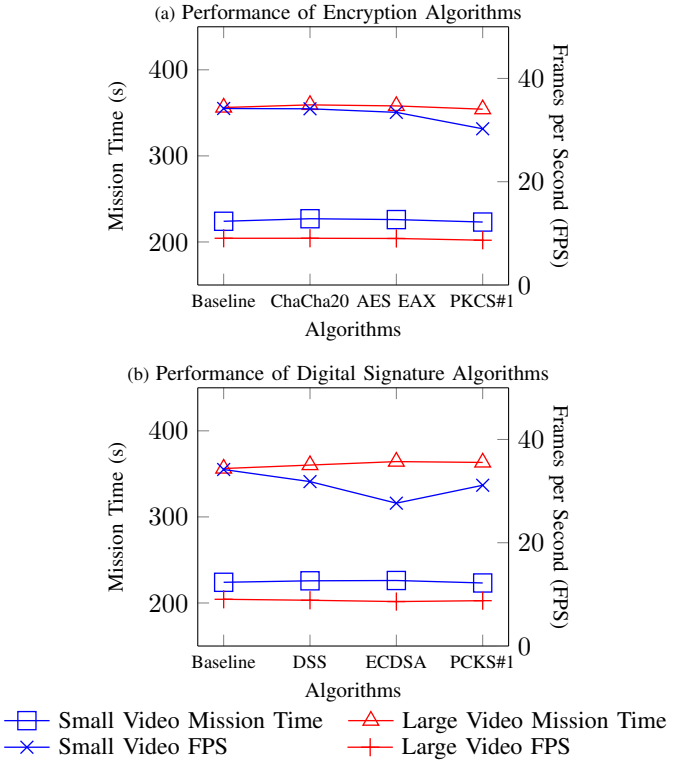


Figure 4: Comparison of performance results between Encryption and Digital Signature Algorithms in the AERPAW testbed.

continuous authentication using digital signatures. To overcome challenges posed by unbalanced datasets in machine learning models for attack detection, we utilized Generative Adversarial Networks (GAN) for creating balanced datasets, as well as improving the detection of sophisticated cyber threats. Our experimental results demonstrate the efficacy of these security measures, with a comparative analysis considering packet encryption, digital signatures, and nonces in secure MAVLink-based communication. Future work includes exploring additional attack vectors, further investigations on machine learning for enhanced security, and addressing system scalability for larger drone networks and complex mission scenarios.

References

[1] M. Ghamari, P. Rangel, M. Mehrubeoglu, G. S. Tewolde, and R. S. Sherratt, "Unmanned aerial vehicle communications for civil applications: A review," *IEEE Access*, vol. 10, pp. 102492–102531, 2022.

[2] A. E. Morel *et al.*, "Enhancing network-edge connectivity and computation security in drone video analytics," in *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–12, IEEE, 2020.

[3] N. Dilshad, J. Hwang, J. Song, and N. Sung, "Applications and challenges in video surveillance via drone: A brief survey," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 728–732, IEEE, 2020.

[4] A. E. Omolara, M. Alawida, and O. I. Abiodun, "Drone cybersecurity issues, solutions, trend insights and future perspectives: a survey," *Neural Computing and Applications*, vol. 35, no. 31, pp. 23063–23101, 2023.

[5] V. Marojevic *et al.*, "Advanced wireless for unmanned aerial systems: 5g standardization, research challenges, and aerpaw architecture," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 22–30, 2020.

[6] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[7] G. Huang and A. H. Jafari, "Enhanced balancing gan: Minority-class image generation," *Neural computing and applications*, vol. 35, no. 7, pp. 5145–5154, 2023.

[8] "Mavlink: Micro air vehicle communication protocol." https://mavlink.io/. Accessed: December 23, 2023.

[9] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro air vehicle link (mavlink) in a nutshell: A survey," *IEEE Access*, vol. 7, pp. 87658–87680, 2019.

[10] R. Chin, C. Catal, and A. Kassahun, "Plant disease detection using drones in precision agriculture," *Precision Agriculture*, pp. 1–20, 2023.

[11] K. AL-Dosari, Z. Hunaiti, and W. Balachandran, "Systematic review on civilian drones in safety and security applications," *Drones*, vol. 7, no. 3, p. 210, 2023.

[12] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet of Things*, vol. 11, p. 100218, 2020.

[13] A. Grote, E. Lyons, K. Thareja, G. Papadimitriou, E. Deelman, A. Mandal, P. Calyam, and M. Zink, "Flypaw: Optimized route planning for scientific uavmissions," in *2023 IEEE 19th International Conference on e-Science (e-Science)*, pp. 1–10, IEEE, 2023.

[14] "Qgroundcontrol, control and mission planning for any mavlink enabled drone." http://qgroundcontrol.com/. Accessed: December 23, 2023.

[15] "Ardupilot, a trusted, versatile, and open source autopilot system supporting." https://ardupilot.org/. Accessed: December 23, 2023.

[16] S. Bagui, D. Mink, S. Bagui, S. Subramaniam, and D. Wallace, "Resampling imbalanced network intrusion datasets to identify rare attacks," *Future internet*, vol. 15, no. 4, p. 130, 2023.

[17] A. N. Mahzer, J. Waleed, and A. T. MaoLood, "Developed lightweight cryptographic algorithms for the application of image encryption: A review," *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 13, no. 2, pp. 11–22, 2021.

[18] A. Roy and S. Karforma, "A survey on digital signatures and its applications," *JCIT*, vol. 3, pp. 45–69, 01 2012.