Stop seeing ghosts: UAV broadcast authentication

Thomas Ewert, Thomas Strang, Leonardus J.A. Jansen
Institute of Communications and Navigation
German Aerospace Center (DLR)
Wessling, Germany
{thomas.ewert, thomas.strang, leonardus.jansen}@dlr.de

***Abstract*—Over the last decade, significant research has been conducted in the field of connected vehicles, particularly in the automotive sector via Vehicular Ad hoc Networks (VANETs), but also extending into airspaces using drones or Electrical Vertical Take-off and Landing (eVTOL) vehicles. With numerous applications ranging from delivery services, search & rescue to commercial and private operations, it is expected that the transportation ecosystem will evolve to accommodate increasing levels of traffic and autonomy - particularly in the urban lower airspace with drones playing a vital role.**

Larger number of flights however also increase the risk of collisions proportionally. While some threats can be mitigated through Unmanned aeronautical vehicle Traffic Management (UTM) methods, such as conflict-free flight plans, an independent last line of defense is essential.

DroneCast aims to address this need by providing a collaborative collision avoidance system based on broadcasted position reports of all drones. However, to prevent malicious actors from disrupting orderly traffic flow by injecting ghost targets that cause unnecessary avoidance maneuvers, robust cybersecurity measures are required.

This paper evaluates strategies for securing broadcast communication systems in resource-constrained environments. The analysis is conducted exemplary on DroneCast, but can be transferred to any broadcast system such as VANETs or Wireless Sensor Networks (WSNs). Given the established constraints, different commonly used practices are compared to offer a justified proposal, which can be the foundation for future implementation within DroneCast or similar applications.

***Index Terms*—UAV, Drone, Collision Avoidance, Broadcast Authentication, Digital Signatures, TESLA, MAC**

I. Introduction

In recent years, technologies of connected vehicles have advanced significantly. In the automotive sector, innovations ranged from Cooperative Collision Avoidance systems utilizing Vehicle-to-Everything (V2X) connectivity to Connected Adaptive Cruise Controls via 5G [1]. Similar approaches have been rolled out in the railway domain with the Railway Collision Avoidance System (RCAS) [2]. At the same time, advances in technology have made drones an affordable and accessible means of transport primarily for goods, but eventually also passengers in the near future. Use cases can be found in various industries, including defense, Search and Rescue (SAR), and parcel delivery. Companies like Amazon are actively exploring ways to integrate drones into 'last-mile' delivery operations, and have already started trials, including those conducted in the Phoenix metropolitan area of Arizona [3]. If these ambitious plans take shape, the number of drones over major cities could increase tremendously, and thus the risk of collision [4].

Current manned air travel relies primarily on Air Traffic Control (ATC) and visual separation to prevent mid-air collisions, with Traffic Collision Avoidance System (TCAS) and Automatic Dependent Surveillance - Broadcast (ADS-B) serving as the last line of defense for vertical resolution advisories. While it is expected that UTM will mainly avoid conflicts by pre-planning collision free trajectories and monitoring such through the UTM system, the lack of redundancy would compromise the safety requirements common in any transportation domain. Existing systems from modern airliners could be repurposed for drones, but may introduce inherited vulnerabilities. For example, TCAS relies on coarse-grained distance and altitude data for conflict resolution, eventually advising vertical maneuvers, which may be inadequate in constrained drone airspace. Moreover, both TCAS and ADS-B lack proper cybersecurity measures and are susceptible to known exploits [5], [6]. With the increasing ease with which potential attackers can acquire the necessary capabilities [7], using inherently insecure technologies for massive drone deployment is not advisable.

The minimize the risk of inflight collisions, the German Aerospace Center (DLR) is developing a Drone-to-Drone (D2D) communication and surveillance system, addressing the need for a last-line of defense, within the DroneCast project. The technology is envisioned to serve as TCAS alike collision avoidance for drones in the future urban lower airspace [8]. It is based on periodical beacon messages broadcasted by every drone consisting of position- and speed information, allowing any other drone in reception range to calculate a potential point of collision on their flight paths which in turn can be circumnavigated to avoid the collision happening.

Despite the fact that any of the cooperative anti-collision technologies in road, rail and air is meant to protect the safety of its users, there exist malicious forces trying to compromise those approaches for their very own benefit. For instance, one could try to keep away drones over their own properties by having any drone react and circumnavigate another drone over that property using DroneCast conform beacon messages. Even worse, it is much easier to have an artificial, non-existent but virtual drone transmit those messages than a real one, becoming a ghost target for any real drone.

In the following, we particularly analyze the suitability of different methods to secure broadcast messages in order to pre-

vent the unauthorized insertion of ghost targets, i.e., artificially generated messages intended to masquerade as legitimate transmissions and mislead recipients. While our evaluation applies to various broadcast systems, we use DroneCast and its resource constraints as a practical example. Chapter II therefore provides a brief introduction to the system and its restrictions, followed by an analysis of commonly used cryptographic approaches, each described with these constraints in mind. The paper concludes with a comparison of their suitability in the broadcast environment and offers recommendations for future implementations and open research topics.

II. Background

DroneCast is a communication system under development that provides collaborative detect-and-avoid functionality for collision prevention by broadcasting position and speed reports at regular intervals. It is designed as an independent system with the characteristics of urban mobility in mind and supports a minimum of 100 drones per surface square kilometer. Transmission is foreseen in a 5 MHz channel of the C-band between 5030 MHz and 5091 MHz, which has been allocated for drone communication in many countries. While the minimum reception range required for collision avoidance varies and depends on multiple factors, authors in [8] assume that a minimum system range of 1 km is sufficient for the urban environment.

As DroneCast operates in an area subject to strong multipath propagation, the basic physical architecture follows an Orthogonal Frequency-Division Multiplexing (OFDM) approach as it offers various benefits in this environment. Further, robustness is increased by the application of Dual Carrier Modulation (DCM). To enable operation with a large density of drones, broadcast message duration should be restricted to approximately 1 ms in length and a nominal rate of 1 Hz on average [9]. With providing situational awareness for all drones as the major goal, message content primarily focuses on speed and location information. The corresponding data fields are shown in figure 1. Current drafts also foresee a 32 bit Cyclic Redundancy Check (CRC) for forward error correction. As cybersecurity is of fundamental importance, an additional 256 bit are reserved in every message for security measures, making each broadcast message 484 bits in size [8], [9].

Drones in reception range can utilize the received position and speed vector data to determine the current proximity to the sender and anticipate any potential flight-path conflicts, allowing them to take corrective action if necessary. However, the ability to detect and avoid conflicting drones greatly depends on the ability to receive the corresponding broadcast message. As there is no deterministic access control foreseen for DroneCast, access to the physical layer will be handled by Location Based Time Division Multiple Access (LB-TDMA) through the utilization of location data to minimize message collisions. Hereby, time is divided into frames which are further divided into slots, all being synchronized via Global Navigation Satellite System (GNSS) among the drones. To cover propagation and clock errors, each slot is further split into a transmission and guard part. Each drone should transmit once per frame, but can only randomly select from a group of slots which is determined based on its own current position. Closely located drones therefore transmit on different slots which in turn reduces multiple-access interference [10]. While transmission through selected slots still results in broadcast messages at a 1 Hz rate on average, they are not evenly spaced over time.

Information	Size [bit]
3D Position	48
Position Uncertainty	3
Time	32
Position Source	3
Speed Vector	32
Speed Vector Uncertainty	3
Identity Drone	24
Identity Operator	24
Identity Mission Plan	24
Flight State	3

Fig. 1: Expected DroneCast beacon data with bit sizes [8]

III. Analysis

Any communication system in its applications on top might be prone to a rich set of possible security attacks. This paper focuses on ghost targets as one possible attack vector used to yield collision avoidance noneffective. In order to protect a system from processing non-reliable transmissions, each receiver must be able to distinguish legitimate from illicit ones. Although there are different methods to achieve this, our approach focuses on protecting the integrity and authenticity of messages. This ensures that the receiver can determine the origin, i.e., the indicated Unmanned Aerial Vehicle (UAV), of the transmitted information and that it has not been tampered with on the way between sender and receiver.

However, any chosen algorithm also has to support the additional requirements created through the system. For instance, bandwidth constraints or computational limitations may restrict the use of otherwise widely adopted methods. Based on the description from the previous chapter, we have identified DroneCast's most restrictive characteristics and their implications in Table I. Since the system remains under development, we have also outlined our assumptions as a basis for further analysis.

The following section introduces commonly used concepts and evaluates their suitability with the given parameters.

A. Message Authentication Codes (MACs)

Message Authentication Codes (MACs) can provide authenticity and integrity of data and are based on symmetric cryptography, requiring the recipient and sender to share a secret key before any transmission can be protected [11]. Therefore MACs can be seen as symmetric digital signatures, usually incorporating some kind of hashing of the content of the message to be signed.

TABLE I: Major design factors introduced through DroneCast system

Factor	Implications	Assumption
Bandwidth	Limited bandwidth constrains the choice of cryptographic algorithms and favors a compact output. Larger cryptographic outputs may increase transmission times, necessitate compression or might not be useable at all for the corresponding system.	A maximum of 256 bits, plus 32 bits CRC, is currently allocated for security data in DroneCast.
Transmission Rate	Higher transmission rates require computation and validation to occur within the transmission interval, therefore preventing lag and benefiting real-time responsiveness. Hardware optimization might be required with higher rates.	DroneCast envisions to transmit messages at a rate of 1 Hz.
Number of Recipients	Systems with multiple recipients scale poorly using purely symmetric encryption due to the complexity of key management. Single-recipient systems, such as ATC towers or Road-Side-Unit (RSU), could utilize pairwise shared keys, which are easier to manage. Every recipient needs to have access to at least some information in order to verify messages, while this information has to be updated if changes occur.	Within DroneCast, every drone potentially has to verify all received broadcast messages. With a slot length of 1 ms, up to 1,000 messages can be received per second.

1) Limitation by Bandwidth: In digital aeronautical communications security, MACs are used for instance in the ACARS Message Security (AMS) system [12], [13]. Due to the short message sizes and low throughput of the initial Aircraft Communications Addressing and Reporting System (ACARS) [14], [15], the length of used MACs in AMS is truncated to 32 bits [12]. The probability of an attacker guessing the correct MAC being $\frac{1}{2^{32}}$ combined with the short validity of the key of maximum one flight, the usage of such shortened MACs has been evaluated as being sufficient [12]. While the German Federal Office for Information Security (BSI) recommends a minimum length of 96 bits, National Institute of Standards and Technology (NIST) is giving further guidance on the required length for MAC tags. Assuming that guessing is the attacker's only option, the probability of a randomly generated MAC being accepted as valid is $1/2^{\text{length}}$. A minimum MAC length of 64 bits is recommended if the protocol design does not limit the number of failed verification attempts before retiring the key. Otherwise, the MAC length should satisfy the inequality:

$$\text{length} \geq \log_2 \left(\frac{\text{MaxInvalids}}{\text{Risk}} \right)$$

where *Risk* specifies the maximum acceptable probability of an inauthentic message being mistakenly accepted as valid (e.g., 2^{-20}) [16].

2) Limitation by Transmission Rate: Symmetric cryptography is generally faster than its asymetric counterpart. Performance optimized algorithms such as BLAKE2b exist, capable of processing one byte per approximately 3 CPU cycles [17]. For example, on an ARM Cortex M3, as it is used in the Arduino platform, clocking at 72 MHz, one 484 bits long message could be hashed within 2.54 microseconds. NIST has further standardized the ASCON algorithm as its lightweight cryptography solution, optimizing for usage on resources restricted devices such as IoT [18]. While benchmarks show, that ASCON-hash is approximately half the speed than BLAKE2, hasing could still occur within 5.3 microseconds on a platform such as Arduino [19]. It can further be assumed, that final products will use hardware accelerators for such calculations, further reducing the time needed. The introduced algorithms could be used for e.g., a Keyed Message Authentication Code (KMAC) generation, with processing time well below the transmission intervals.

3) Limitation by Recipient Number: While being efficient due to their symmetric nature, MACs also requires that all message receivers and senders have knowledge of the key used for its generation. For classical tags, authenticity cannot be established if the key is known to more than two communication partners. For the required one to one communication between n drones, this would require the exchange of $(n(n-1))/2$ keys and the transmission of (n-1) individually secured broadcast per drone, making the system not scalable nor efficient. This limitation is inherit to many symmetric cryptography based algorithms, and would also apply if, e.g., encryption has been chosen to secure broadcast messages.

B. Asymmetric Digital Signatures

A very common cryptographic method to verify the origin and integrity of a message are asymmetric digital signatures, using a private key to sign the hash of a message which every receiver can verify using he matching public key of the sender. In order to achieve security comparable to symmetric cryptography algorithms, public key algorithms require longer operands and therefore are up to 2-3 times slower than symmetric algorithms [11].

1) Limitation by Bandwidth: The resulting signature sizes are not negligible if the 128 bit security strength recommended by the NIST for usages beyond the year 2031 should be achieved. For instance, RSA-3072 - providing roughly the same security as a 128-bit symmetric key by requiring about 2^{128} operations to break - computes a 3072 bit long signature.

Currently, NIST has approved three families of digital signature algorithms, Rivest Shamir Adleman (RSA), Elliptic Curve Digital Signature Algorithm (ECDSA), and Edwards Curve Digital Signature Algorithm (EdDSA). Smallest signatures are created by the latter two, with ECDSA-256 based on P-256 elliptic curves being 512 bits in size while still providing a 128-bit security strength. However, algorithms with shorter signature lengths have been proposed, such as the Boneh-

Lynn-Shacham (BLS) scheme used in Ethereum 2.0 offering signatures of approximately 384 bits in length. Current public key schemes are vulnerable to potential breakthroughs in quantum computing, prompting a focus on developing quantum-safe algorithms. Various proposals are either advancing through the NIST standardization process or have already been standardized. However, a common characteristic of post-quantum algorithms is their (significantly) larger signature sizes compared to traditional schemes. Among standardized NIST post-quantum algorithms, Falcon offers the shortest signature at 666 bytes (5,328 bits). From the algorithms still undergoing the standardization process, UOV Is-pkc achieves the shortest length, at 96 bytes (768 bits) [20].

Application of signatures for size-constraint broadcasting differs depending on whether the generated output exceeds the available space in each message, or if the content needs to be split and transmitted via multiple ones.

a) Space-Compliant Signatures for DroneCast: If algorithms become available providing a signature length of maximum 256 bits, the output could fit into one broadcast message. This would presuppose, that no signature encoding such as Distinguished Encoding Rules (DER) has to be considered and the raw bits can be attached directly within the standardized message structure. Verification could happen immediately at the receiving party, assuming the sender's public key is known. While the use of a CRC could be seen redundant in this case, it could be kept in order to avoid processing of incorrectly received messages. However, if needed, the additional 32 bit could be utilized if required to fit one signature into one message.

b) Space-Exceeding Signatures for DroneCast: If the chosen algorithm generates signatures longer than 256 (+ 32) bits, broadcasting them will require multiple messages. Ideally, the receiver should know which signature segment each message carries. This can be done using marker bits (e.g., from reserved CRC space) or time-based methods—such as using frame numbers or timestamps in synchronized systems like DroneCast. If transmissions are irregular or involve multiple segments, marker bits are likely required. Alternatively, the receiver can reconstruct the signature by combining recent messages in a trial-and-error manner, if resources allow. As the number of required messages increases, the proportion of verifiable broadcast transmissions decreases, since only the first message in a sequence can be used to generate the signature. For instance, with a 512-bit signature and no additional encoding, only every second message could be authenticated. The adverse effects could be reduced as each message also includes a position vector, whose information can be used to verify the plausibility of non-authenticated position reports.

2) Limitation by Transmission Rate: Asymmetric cryptography includes more resource intensive calculations and therefore requires better hardware or longer processing times. Software implementations show ECDSA at roughly 350 ms for signing and verification, while BLS achieves 108 ms / 166 ms on an Intel i5 CPU [21]. Hardware accelerators like Trusted Processing Modules (TPMs) or Field Programmable Gate Arrays (FPGAs) can improve this further; for example, ECDSA-256 reaches 7.15 ms for signing and 9.09 ms for verification [22], [23], with potential for further reduction on more capable hardware. While a single verification is possible, verifying 1,000 messages per second would take 9.09 s—exceeding real-time constraints and causing a growing backlog that could overwhelm the system. A trade-off may be necessary, either by upgrading hardware or selectively verifying only relevant messages, such as those from drones in close proximity to the receiver.

3) Limitation by Recipient Number: As is standard with any asymmetric cryptography, the corresponding public keys must be provided to the receiver to enable verification and ensure the authenticity of the sender's identity. This is typically achieved through certificates, which are issued digitally signed by a trusted Certificate Authority (CA) and distributed as part of a Public Key Infrastructure (PKI). Given that drones operate over smaller geographic regions compared to aircraft, the certificate issuing authority could be more localized. For example, this authority could be the Civil Aviation Authority (CAA) or Air Traffic Services (ATS) provider of the respective country or regional oversight bodies such as European Union Aviation Safety Agency (EASA) in Europe.

Certificate sizes vary depending on the algorithm and chosen format. In the commonly used X.509 format, ECDSA-256 public key certificates typically require around 352 bytes when including the minimum necessary fields [24]. However, further adjustments may be required to tailor the certificate structure for the specific needs of the UAV use case. While the following gives a short introduction in different ways of certificate distribution, the exact procedure is outside the scope of this paper.

In general, distributing certificates could occur in an online or offline fashion. In the latter case, an external data connection is required before each flight to ensure the issued certificates are available to the UAV. However, any certificates issued afterward cannot be included for the duration of the flight. This problem could be mitigated by, for example, enforcing a minimum time interval between certificate issuance and the UAV's first flight. If the interval exceeds the maximum flight time of any UAV, the availability of all required certificates at take-off time can be ensured. If flight time is too long or the possibility for offline updates are too infrequent, certificate distribution can be achieved via the broadcast system itself. The security data portion or even the entire messages could be replaced with certificate information. The required number of transmissions varies depending on the approach. Messages could be sent sequentially, allowing for a fast transmission but temporarily preventing position reports or their verification. Alternatively, they could be distributed over time, slowing the transmission but minimizing system functionality outages.

Digital signatures not requiring the distribution of certificates have also been proposed by Shamir in 1984. Identity Based Signature (IBS) utilize the ID of the sender to derive the public key and verify the signature. The same principle

has been applied to BLS signatures and generate an output of 384 bits in length [25]. Other applications in the area of VANETs using Cha Cheon's ID Based Signatures with a size of 464 bits have been explored in [26]. One disadvantage that all IBS schemes have in common is the inherent key escrow issue, as the Key Generation Center (KGC) holds the unique private keys for each participants. While further research and risk assessments might show the suitability of this solution, it has not been considered within this paper.

C. Timed Efficient Stream Loss-tolerant Authentication (TESLA)

A potential solution to still leverage the performance benefits of MACs is TESLA, which is guaranteeing authenticity by transferring validation to the time domain, which is already used in the transport domain e.g. [27] [28]. Hereby, the sender generates a hash chain and reversely uses the created values as keys for the MAC tag generation. In each message the last used value from the hash chain is revealed. The receiver can then verify the previously received tag and also check if the used key is part of the senders hash chain. This requires, that the receiver has knowledge of

- the beginning of the reversely traversed hash chain (which can be calculated from any received key element)
- the owner of the key chain
- the timestamp of the beginning value and
- the key release interval length.

This approach remains effective even if some messages are not received, which can be assumed in not deterministic broadcast systems.

1) Limitation by Bandwidth: Message integrity is checked via the utilization of MACs, while authenticity is guaranteed via the delayed key release. With the amount of failed verification attempts not be fully controllable, as it is done by each drone independently, the minimum advised MAC length of 64 bits will be used [16]. Two main parameters within TESLA are the key release delay and interval. Both influence the verification delay directly, meaning a longer interval leads to a larger delay. The fastest key release would happen within the next broadcast message. With currently just 64 bits occupied, the remainder 192 bits would be sufficient to hold an e.g., 128 bit symmetric key within one broadcast message. Note: With a key release within the next transmission frame, a verification delay of worst case two second is introduced.

2) Limitation by Transmission Rate: Performance of the required calculations can be assumed similar to the ones in section MACs. From a computational point of view, the storage requirements should be highlighted. Prior to utilizing TESLA, an entire key chain has to be pre-calculated and stored at each sender. Assuming a two-hour operational window is sufficient for a drone, a key chain operating at a rate of 1 Hz would require over 7,200 keys, equating to approximately 113 KB of storage. Depending on the hardware security requirements, such keys can be stored either within a TPM, provided sufficient space is available, or externally on the drone's storage in a key-wrapped format. The latter approach

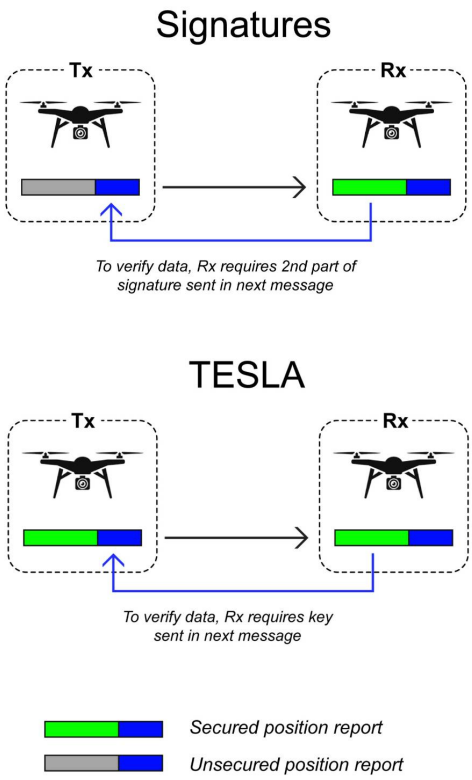

Fig. 2: Comparision of delay caused by Signatures and TESLA

introduces additional latency due to the need for unwrapping keys before they can be used. Note that every key in the pre-calculated chain is assigned to a specific time slot without further measures. Even if the drone is not airborne at this time, keys will have to be skipped in order to prevent re-use of keys recorded by a malicious entity. In order to reduce the amount of keys to be pre-computed and stored, the validity interval per key could be extended. Similarily, the verification delay increases as well. An optimum between hardware requirements and verification delay has to be found here. Too long validity intervals would be delaying the process too much, as unverified messages should not be used for collision avoidance.

3) Limitation by Recipient Number: As previously mentioned, any receiver would require knowledge about the key chain's owner and validity timestamps in order to verify incoming messages. Such information could be packaged into a certificate and then distributed to all communication partners. The same effects as described in the certificate distribution section of digital signatures would apply initially. However, the operational timestamp of TESLA correlates with the length of the pre-calculated key chain. If it runs out during flight, no more broadcast message can be secured, unless a new chain is calculated.

TABLE II: Overview of difference between TESLA and signature-based approaches

Overhead type	TESLA	Digital Signatures
Storage	Pre-computed key chain (113 KB for 2 hours) and at least one verified key with timestamp for each monitored drone. Optional: own certificate for end key of chain (if online certificate distribution).	Private key, public keys of monitored drones or certificates (352 bytes per certificate in X.509 format). If online distribution, own certificate is needed.
Transmission	64-96 bit MAC tag and 128-bit released key, transmitted in every message.	256-512 bit signatures; single-message or multi-message transmission depending on signature length. Certificates require 12+ messages to transmit (online distribution).
Computation	Lightweight computation for MAC verification. Requires pre-calculation of key chains. Delayed key release adds minimal runtime cost.	Digital signature verification is computationally intensive. Certificate handling adds to overhead, especially for online distribution.
Delay	Verification delay depends on key release interval; minimum 1 second with 1 Hz frequency.	Delay due to signature verification and certificate retrieval (if online). Multi-message signatures increase delay proportionally, with a two step transmission verification delay of one second with every other message verified.

IV. Evaluation

While all proposed approaches introduce varying levels of security overhead to DroneCast, the exact type and extent depend on the chosen parameters and algorithm. In general, these overheads can be assigned to the categories storage, computational, or delay.

At their core, all approaches face the same challenge of providing the communication partners with the required cryptographic information to verify broadcast messages. For a purely MAC based solution, this presents a bottleneck, excluding this approach from further considerations.

For both, digital signatures and TESLA, this information is assumed to be distributed in the form of a certificate. Regardless of the chosen method, the initial overhead is going to be similar. However, unlike conventional public key certificates, which remain valid until their expiration date unless revoked, TESLA certificates (can be seen as an asymmetric signature of a specific chain) are limited by the length of their precomputed key chain. Consequently, a sufficient chain length has to be ensured to avoid flight time limitations.

Both solutions introduce a verification delay of at least one broadcast interval, depending on the selected transmission rate. The signature approach, assuming n messages are required to transmit a whole signature, further can only authenticate $1/n$ broadcast messages, an effect that can be softened by utilizing the trend vector of each transmission. A visual comparison of both approaches is shown in 2. However, verification of asymmetric signatures also requires higher computational power and therefore time than that of MAC tags, leading to a possible backlog or the need to only selectively verify incoming messages.

In regards to system complexity the two approaches differ, with digital signatures being the simpler option as no need for key chain preparation is given.

The real strength of TESLA becomes particularly significant in two cases. On the one hand, when signature sizes vastly exceed the available space, as is the case with post-quantum signatures. As symmetric cryptography is not as affected by advances in quantum computation, TESLA would present a suitable solution in this environment. On the other hand, TESLA is also less susceptible to message losses. While DroneCast aims at minimizing the occurence of overlapping transmissions, they can not completely be prevented in a broadcast system. If interference renders a message unreadable, TESLA can utilize the key material transmitted in subsequent intervals to compute the missing keys using the hash function and then verify received broadcast messages. In the signature-based approach, all messages containing parts of the same signature must be received completely and integer for verification. If even one bit is missing/flipped, verification is delayed until the next complete set of signature messages has been received.

Table III summarizes the mentioned key properties and highlights which approach offers the most favorable impact on each.

Property	Digital Signatures	TESLA
Latency	-	-
Computation	- -	+
Storage	+	-
Complexity	+	-
Packet Loss Resilience	-	++
Per-message Verification	-	+

TABLE III: Pros and Cons across key properties

In the summary above, we have placed particular emphasis on resilience to packet loss, given its frequent occurrence in broadcast systems, as well as the ability to perform required verifications in a timely manner.

Our computational analysis compared the susceptibility to message loss between both approaches by evaluating the average time between two verifiable messages. Figure 3 illustrates this average delay as a function of the message loss probability given different distributions. The results show that, particularly under high message loss conditions, signature-based schemes introduce significant delays due to the decreased likelihood of receiving two sequentially linked messages. Although the DroneCast medium access protocol is designed to minimize collisions and thus maintain a low message loss rate, TESLA demonstrates greater resilience and maintains better performance as conditions degrade.

We therefore recommend the use of TESLA for ensuring broadcast message authentication and integrity for the

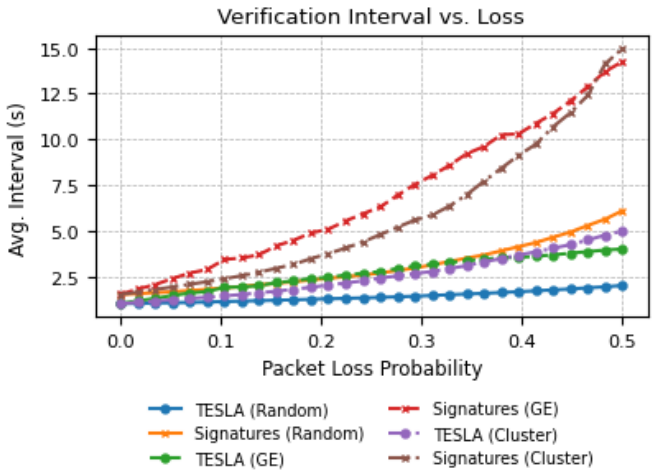

Fig. 3: Average interval between two verifiable messages using TESLA and digital signatures under different packet loss models. Initial values (1s for TESLA, 1.5s for signatures) reflect the number of packets required before two messages become verifiable: TESLA requires three, while signatures require four. The Gilbert-Elliott (GE) model represents bursty losses by alternating between a low-loss and high-loss state, while the clustered model simulates loss bursts using random-length clusters.

DroneCast and similar use cases. This approach also has the advantage that system characteristics remain unchanged if post-quantum cryptography needs to be adopted. To distribute the necessary information to all drones, we define three operating modes: offline, where all data is provided before takeoff—particularly important for large post-quantum certificates; online, where some messages are unauthenticated or replaced by certificate data; and mixed, which combines both approaches.

Even with TESLA in place, it is important to address how unauthenticated messages are handled for collision avoidance. A legitimate UAV may fail verification due to factors such as missing information, short authentication windows, or high message loss rates. Therefore, appropriate procedures should be established to manage such cases. One possible approach is a point-based trust system, where each unauthenticated message is assigned a penalty score. If the accumulated score for a sender exceeds a defined threshold, subsequent messages can be treated as potentially fraudulent and discarded.

Below is a summary of how penalty points could be assigned to each unauthenticated message. In this system, each point is negative and reduces the sender's overall trust level. In order to reduce processing overhead, a recipient could filter out messages of UAVs too far away or with travel paths not conflicting with its own.

Apply trust penalties if:

- Claimed location does not align with the reception duration — drones closer to the receiver should be consistently receivable, whereas distant drones may have intermittent reception.
- The receiver has directional antennas and signal direction does not align with the claimed position.
- The UAV reports the same position over an extended period of time, therefore blocking a certain area.
- The reported positions do not match expected movement trends — e.g., sudden direction changes, jittering, or unrealistic flight patterns.

While the outlined point-based penalties serve as an example, they can be adjusted to suit local operational requirements. Additionally, the threshold for discarding messages should be defined in accordance with safety and regulatory standards. Instead of—or in addition to—using fixed rules, a lightweight machine learning (ML) model could be trained to dynamically learn from past data.

V. Conclusions

Within this paper, we have shown an approach on how to find a security solution to ensure authenticity and integrity protection for resource constraint cooperative broadcast systems becoming popular in maritime, road, rail and air transport systems. We have used the collision avoidance system for UAVs and its restrictions as a practical example and highlighted different approaches for broadcast authentication based on the system's limitations. While each approach has its merits, in environments where transmission loss is unavoidable, tolerance to loss becomes critical. TESLA emerged as the optimal solution, primarily due to its robust performance under such conditions. While drawbacks such as the need for pre-calculated key chains exist, we found that the benefits outweigh them. Further research should focus on optimizing the distribution of required certificates, as this aspect was beyond the scope of this work. Additionally, further work should refine the exact protocol specifications and evaluate its performance in flight trials.

Acronyms

ADS-B	Automatic Dependent Surveillance - Broadcast
AMS	ACARS Message Security
ATC	Air Traffic Control
ATS	Air Traffic Services
BLS	Boneh-Lynn-Shacham
BSI	Federal Office for Information Security
CA	Certificate Authority
CAA	Civil Aviation Authority
CRC	Cyclic Redundancy Check
DER	Distinguished Encoding Rules
DLR	German Aerospace Center
D2D	Drone-to-Drone
DCM	Dual Carrier Modulation
EASA	European Union Aviation Safety Agency
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards Curve Digital Signature Algorithm
eVTOL	Electrical Vertical Take-off and Landing
FPGA	Field Programmable Gate Array
GNSS	Global Navigation Satellite System

IBS	Identity Based Signature
KGC	Key Generation Center
KMAC	Keyed Message Authentication Code
LB-TDMA	Location Based Time Division Multiple Access
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
OFDM	Orthogonal Frequency-Division Multiplexing
PKI	Public Key Infrastructure
RSA	Rivest Shamir Adleman
RCAS	Railway Collision Avoidance System
RSU	Road-Side-Unit
SAR	Search and Rescue
TCAS	Traffic Collision Avoidance System
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TPM	Trusted Processing Module
UAV	Unmanned Aerial Vehicle
UTM	Unmanned aeronautical vehicle Traffic Management
V2X	Vehicle-to-Everything
VANET	Vehicular Ad hoc Network
WSN	Wireless Sensor Network

REFERENCES

[1] S. Y. Gelbal, S. Zhu, G. A. Anantharaman, B. A. Guvenc, and L. Guvenc, "Cooperative collision avoidance in a connected vehicle environment," 2023. [Online]. Available: https://arxiv.org/abs/2306.01889

[2] A. Lehner, T. Strang, I. Rashdan, O. Heirich, B. Siebler, F. Ponte Müller, and S. Sand, "Virtual infrastructure protecting trains on collision course and beyond - virtuelle infrastruktur sichert züge auf kollisionskurs und darüber hinaus," *ZEVrail - Zeitschrift für das gesamte System Bahn*, pp. 189–195, 2017. [Online]. Available: https://elib.dlr.de/119952/

[3] Tyler Greenawalt, "Amazon has launched our most advanced delivery drone yet—here's everything you need to know," https://www.aboutamazon.com/news/operations/mk30-drone-amazon-delivery-packages(accessed December 18, 2024), About Amazon, Tech. Rep., 12 2024.

[4] M. Doole, J. Ellerbroek, and J. Hoekstra, "Drone delivery: Urban airspace traffic density estimation," in *8th SESAR Innovation Days, 2018*, 2018, sIDs2018: 8th SESAR Innovation Days, SIDs2018 ; Conference date: 03-12-2018 Through 07-12-2018.

[5] G. Longo, M. Strohmeier, E. Russo, A. Merlo, and V. Lenders, "On a collision course: Unveiling wireless attacks to the aircraft traffic collision avoidance system (TCAS)," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 6131–6147. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/longo

[6] A. Costin, A. Francillon *et al.*, "Ghost in the air (traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices," *black hat USA*, vol. 1, pp. 1–12, 2012.

[7] T. Ewert and N. Mäurer, "Safety and security considerations on the airbus wake energy retrieval program "fello'fly"," 04 2023, pp. 1–12.

[8] L. M. Schalk and D. Becker, "Dronecast – analysis of requirements and discussion of first design decisions," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, 2022, pp. 1–9.

[9] D. Becker and L. M. Schalk, "Dronecast - physical layer design and measurement-based simulation analysis for urban drone-to-drone communication scenarios," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, 2024, pp. 1–7.

[10] ——, "Dronecast - mac layer design and optimization," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, 2024.

[11] C. Paar and J. Pelzl, *Understanding cryptography: A textbook for students and practitioners*. Springer Science & Business Media, 2009.

[12] ARINC, "DATALINK SECURITY PART 1 – ACARS MESSAGE SECURITY," Aeronautical Radio, Incorporated (ARINC), Tech. Rep., 12 2007, [Online]. Available: https://standards.globalspec.com/std/1039315/ARINC823P1.

[13] ——, "DATALINK SECURITY PART 2 – KEY MANAGEMENT," Aeronautical Radio, Incorporated (ARINC), Tech. Rep., March 2003, [Online]. Available: https://standards.globalspec.com/std/1039315/ARINC823P1 [Accessed: February 23, 2021].

[14] ——, "Aircraft Communications Addressing and Reporting System," Aeronautical Radio, Incorporated (ARINC), Tech. Rep., February 1998, [Online]. Available: https://web.archive.org/web/20120510105708/https://www.arinc.com/cf/store/catalog_detail.cfm?item_id=561 [Accessed: January 23, 2021].

[15] RTCA, "DO-281C, Minimum Operational Performance Standards (MOPS) for Aircraft VDL Mode 2 Physical Link and Network Layer," Radio Technical Commission for Aeronautics (RTCA), Tech. Rep., September 2018, [Online]. Available: https://www.rtca.org/products/do-281c-electronic/ [Accessed: January 05, 2021].

[16] M. D. C. S. D. I. T. Laboratory, "Recommendation for block cipher modes of operation: The cmac mode for authentication," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST Special Publication (SP) 800-38B, Includes updates as of October 6, 2016, 2013.

[17] B. 2, "Blake2 - fast secure hashing," 2 2017, accessed: 2025-01-30. [Online]. Available: https://blake2.net

[18] M. S. Turan, K. McKay, D. Chang, L. E. Bassham, J. Kang, N. D. Waller, J. M. Kelsey, and D. Hong, "Status report on the final round of the nist lightweight cryptography standardization process," National Institute of Standards and Technology, NIST Interagency or Internal Report (IR) 8454, June 2023. [Online]. Available: https://doi.org/10.6028/NIST.IR.8454

[19] R. Weatherley, "Lightweight cryptography finalists - performance benchmarks," 2023, accessed: January 30, 2025. [Online]. Available: https://rweather.github.io/lwc-finalists/performance.html

[20] Cloudflare, "A look at the latest post-quantum signature standardization candidates," 11 2024, accessed: 2025-01-01. [Online]. Available: https://blog.cloudflare.com/de-de/another-look-at-pq-signatures/

[21] N.-Q. Luc, Q.-T. Do, and M.-H. Le, "Implementation of boneh - lynn - shacham short digital signature scheme using weil bilinear pairing based on supersingular elliptic curves," *Ministry of Science and Technology Vietnam*, vol. 64, no. 4, pp. 03–09, December 2022, accessed: January 30, 2025. [Online]. Available: https://www.researchgate.net/publication/366530582

[22] S. Azam, V. Rožić, and I. Verbauwhede, "Prime field ecdsa signature processing for reconfigurable embedded systems," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1867–1880, 2012. [Online]. Available: https://ieeexplore.ieee.org/document/836460

[23] P. Holzer, L. Franz, and M. Bogner, "Comparison of ecdsa signature verification implementations on bare-metal embedded systems," 2024, accessed: 2025-01-01. [Online]. Available: https://www.cal-tek.eu/proceedings/i3m/2024/emss/005/pdf.pdf

[24] T. Ewert, N. Mäurer, and T. Gräupl, "Improving usable ldacs data rate via certificate validity optimization," in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, 2022, pp. 1–9.

[25] A. Saxena, "Extending the bls scheme to identity based signatures," 06 2006.

[26] B. Jinila and K. Karuppanan, "An efficient authentication scheme for vanet using cha cheon's id based signatures," *Indian Journal of Applied Research*, vol. 4, pp. 106–109, 10 2011.

[27] N. Mäurer, M. Caamano Albuerne, D. Gerbeth, T. Gräupl, and C. Schmitt, "A secure broadcast service for ldacs with application to secure gbas," in *40th AIAA/IEEE Digital Avionics Systems Conference, DASC 2021*, Oktober 2021, pp. 1–10. [Online]. Available: https://elib.dlr.de/142726/

[28] S. Gewies and T. Strang, "Authentication of the medium frequency r-mode navigation message," in *4th European Workshop on Maritime Systems, Resilience and Security 2024 (MARESEC 24)*, ser. Proceedings of the MARESEC 2024. Zendoo, November 2024. [Online]. Available: https://elib.dlr.de/210067/