Towards message authentication and integrity for COSPAS-SARSAT 406 MHz distress beacons using lightweight ECDSA digital signatures

Ahsan Saleem, Andrei Costin, Hannu Turtiainen, Timo Hämäläinen
Faculty of Information Technology
University of Jyväskylä, Finland
{saleemay,ancostin,turthzu,timoh}@jyu.fi

***Abstract*—COSPAS-SARSAT is a satellite radio location system for aviation, maritime, and land travellers designed to aid search and rescue (SAR) services in distress. This system effectively detects, processes, and relays distress signals, facilitating prompt responses from SAR services. However, COSPAS-SARSAT 406 MHz protocols, both from an architectural and implementation point of view, exhibit fundamental cybersecurity weaknesses that make them an easy target for potential attackers. The two fundamental flaws of these protocols are the lack of digital signatures (i.e., integrity and authenticity) and encryption (i.e., confidentiality and privacy). The risks associated with these and other weaknesses have been repeatedly demonstrated by ethical cybersecurity researchers.**

In this paper, we first present an overview of the insecure design of COSPAS-SARSAT messaging protocols. Subsequently, we propose a lightweight ECDSA *message integrity and authenticity* scheme that works seamlessly for COSPAS-SARSAT 406 MHz protocols. We propose that the scheme can be added as a backward-compatible software-only upgrade to existing systems without requiring expensive architectural redesign, upgrades, and retrofitting. The preliminary implementation, tests, and results from the lab show that our scheme is effective and efficient in adding message authenticity and integrity and represents a promising applied research direction for a low-cost, potentially backward-compatible upgrade for already deployed and operational systems.

I. Introduction

The exponential increase in digitalization and groundbreaking technological advances in most fields and verticals of human lives has brought both advantages (e.g., real-time and broadband connectivity) and novel challenges (e.g., cybersecurity attacks and resilience) to critical domains, such as space, aviation, maritime, and SAR [1], [2], [3], [4], [5]. Space technology is an essential backbone for supporting services that shape daily life, providing a wide range of vital services such as Internet connectivity, media broadcasts, Earth observation data, and global positioning services. Space technology's importance is expected to increase in the future [6]. As the demand for space assets increases, cybersecurity threats also increase. Satellite communication is subjected to diverse cyber-attacks, including eavesdropping, signal injection, and signal spoofing, all of which have the potential to lead to system failures [7], [8]. Several security measures have been proposed to overcome these cybersecurity threats to satellite communication [9], [10], [11], [12].

Space systems play a crucial role in the Search and Rescue SAR domain, with the COSPAS-SARSAT 406 MHz distress beacon serving as a notable example. This system utilizes the Satellite Aided Tracking System (SARSAT), which can be activated during emergencies [13], [14], [15]. This distress beacon alerts the SAR network designed to act on these alerts and dispatches appropriate rescue services. COSPAS-SARSAT is classified into three types of beacons: personal locator beacon (PLB) [16], emergency locator transmitter (ELT) [17], and emergency position indicating radio beacon (EPIRB) [18]. ELTs serve aviation, EPIRBs work in maritime, and PLBs assist in personal uses, and all transmit distress signals during emergencies. These beacons send a 406 MHz digital distress signal containing a unique 15-digit identifier, and the SAR rescue center receives these signals from the COSPAS-SARSAT satellite. Distress signals are decoded to obtain the serial number, location, type of distress, and other data. The COSPAS-SARSAT 406 MHz protocols are inherently insecure and lack confidentiality, message authentication, and data integrity, making them vulnerable to various types of attacks, such as eavesdropping, spoofing, and replay attacks [19].

Compared to aviation (ADS-B) [20] and maritime (AIS) domains [21], to the best of our knowledge, within the COSPAS and SAR field no studies before SpaceSec24 have proactively proposed security schemes to enhance the message authenticity, integrity, and confidentiality of COSPAS-SARSAT protocols. In this paper, we propose a generic and potentially extensible approach that adds cryptographically strong message authenticity and integrity to the COSPAS-SARSAT message communication. The signed messages are seamlessly transmitted in follow-up messages without modifying the existing COSPAS-SARSAT protocols. The proposed scheme aims in principle to be backward-compatible and requires no hardware changes or modifications to the existing protocols and transponders, making it potentially suitable for real-world deployment using software-only upgrades.

Workshop on Security of Space and Satellite Systems (SpaceSec) 2024
1 March 2024, San Diego, CA, USA
ISBN 979-8-9894372-1-4
https://dx.doi.org/10.14722/spacesec.2024.23088
www.ndss-symposium.org

The main contributions of this work are as follows:

1) We propose a software-only approach to adding strong cryptographic support for message integrity, authentication, and anti-replay to existing **non-"secure by default"** COSPAS-SARSAT communications.
2) We propose and demonstrate the *(quasi-)first*[1] message authenticity, data integrity and anti-replay scheme for COSPAS-SARSAT 406 MHz protocols.
3) We provide a security analysis of the proposed scheme, which aims to ensure that the proposed scheme is secured under a well-defined security threat model.
4) We demonstrate and evaluate our approach through a small-scale implementation and simulation[2] and discuss some key takeaways for future research and applications. To allow further community improvements and evaluations, we open source our implementation: https://github.com/Ahsan8/dump406

The rest of the paper is organised as follows. We present the main related works in section II. In section III, we present the relevant background knowledge, system models, and security goals. We further detail our proposed authentication scheme and experimental setup in section IV. In section IV-C, we present a high-level security analysis model for the scheme. We present an evaluation and results of the proposed scheme in section V. We follow this with short a discussion in section VI. Finally, we conclude the paper with section VII.

II. Related Work

This section summarizes the schemes previously proposed for satellite communication security and SAR applications.

Recently, Costin et al. [19] implemented and demonstrated the first attacks on COSPAS-SARSAT 406 MHz protocols. Specifically, they showed that replay, spoofing, and subsequent protocol and application level fuzzing could be launched on these protocols. This study also identifies security weaknesses and possible solutions for enhancing the security of COSPAS-SARSAT. Therefore, securing communication at the protocol level is imperative to ensure the safe and secure operation of the entire COSPAS-SARSAT ecosystem. To counter the attacks outlined by Costin et al. [19], besides our present paper, Khandker et al. [22] also proposed a message authentication, integrity, and anti-replay scheme. The key difference to our present work is that the authors used a message authentication code (MAC) approach using a "shared secret key", which is the same across all communication nodes. Their approach slightly minimizes the overall communication overhead but suffers from the challenges of "shared secret key" approaches, namely, the high risk of leaking the key intentionally or unintentionally, the revocation of keys, and the subsequent reissue and reprogramming of new keys. Moreover, our public–private key approach makes it possible to selectively reject messages from transmitters with compromised keys or suspicious activity based on their private key identifiers.

[1]During SpaceSec24 proceedings, it was revealed that Khandker et al. [22] tackled the same problem with an alternative cryptographic approach.

[2]https://youtu.be/_OlcvkEJylw

Yue et al. [23] showed that both passive security attacks, such as eavesdropping, satellite transponder stealing, and privacy disclosure attacks, and active security attacks, such as spoofing, denial of service (DoS), message modification, and jamming attacks are possible in LEO satellite communication systems (SCSs). Yuqi et al. [24] investigated public walkie-talkie interferences with the COSPAS-SARSAT. In China, showing that public walkie-talkies interfere with the MEO uplink of the COSPAS-SARSAT. Pedersen et al. [25] examined potential security issues in GEO satellite communications. They performed a risk analysis of communication data collected from satellites based on NIST SP 800-30 [26]. The risk assessment revealed 15 threat actors that can lead to various security issues in satellite communication. The analysis showed that with a minimum hardware cost, it is relatively easy to intercept satellite communications and obtain useful information if the information is not protected or insecure protocols are used. Jiang et al. [27] investigated the security issues in secure handoffs, key management, secure transmission control, and secure routing of space information networks. Additionally, they analyzed key management challenges in space information networks, categorizing them as centralized, distributed, topology-based, and preconfigured. Manulis et al. [28] analyzed previously identified security threats and incidents involving satellites to assess the motivations of attackers and the characteristics of adversarial threats. They also provided an analysis of the New Space era, highlighting key technologies and emerging security challenges that influence advancements and innovations in the space and satellite industries. Pavur et al. [29] performed a security analysis and identified previously unknown vulnerabilities in satellites that impact the security and privacy of millions of customers. They also identified the underlying reasons for these security vulnerabilities. In their experiments, they demonstrated that eavesdropping can be performed over satellite communication using inexpensive hardware.

Some studies have analyzed cybersecurity in SAR operations. Stavrinos et al. [30] evaluated the secure operation and interoperability of unmanned underwater vehicles (UUVs) for search, rescue, and military applications. Lechner et al. [31] analyzed cybersecurity risks and attacks in critical infrastructure. They discussed security in a coordination center as a case study to demonstrate the impact of cyber security in SAR services. They provided preliminary work on semantic modelling and simulation to help make critical decisions in SAR and military operations while handling cybersecurity and interpretability issues. Alpiste et al. [32] presented an SAR case study in which they utilized an unmanned aerial vehicle (UAV) and a smartphone with a machine-learning-based object detection mechanism. The proposed model establishes secure communication between the UAV and the smartphone and is currently in use by the Scotland Police for SAR operations. Bernsmed et al. [33] presented a multi-model maritime communication solution using an automatic identification system (AIS) and very high-frequency data-exchange systems (VDES). Their solution used the coordi-

nation of SAR operations as an example and several means of communication, such as different types of networks or transmission technologies.

III. Preliminaries, Models and Goals

This section provides background knowledge and defines our COSPAS-SARSAT model as well as the security goals of the proposed scheme.

A. COSPAS-SARSAT Overview

COSPAS-SARSAT is a satellite-based system established by the U.S., Russia, Canada, and France in 1979 for aviation, marine, and land travellers that detects and locates distress/emergency beacons to support SAR services [34], [35]. During an emergency, the COSPAS-SARSAT beacon is triggered and starts transmitting a 406 MHz distress signal. The COSPAS-SARSAT 406 MHz beacons remain inactive until activated during an emergency or under certain conditions by the user. In maritime cases, for example, all EPIRB beacons float and send out a continuous distress signal for a minimum of 48 hours.

Fig. 1 depicts the typical model of COSPAS-SARSAT 406 MHz operation. Distress signals transmitted at 406 MHz are detected by satellites in the COSPAS-SARSAT network. The COSPAS-SARSAT then passes this information to the mission control center (MCC), which receives and decodes the incoming distress signals, obtains information (e.g., identification, position, country, type of emergency), and then forwards alerts to designated SAR points (e.g., rescue control center, or RCC) as well as other MCCs. The COSPAS-SARSAT system integrates the following three types of satellites: The Low-Altitude Earth Orbit Search and Rescue (LEOSAR) legacy system whose first payload was deployed in 1982; The Geostationary Earth Orbit Search and Rescue (GEOSAR) system, whose first payload was deployed in the mid-late 90s; Medium-Altitude Earth Orbit Search and Rescue (MEOSAR) system, the first payload (Galileo) of which was deployed in 2012 [36], [37], [38]. More than 50 MEOSAR payloads are operational on the Galileo, GPS, GLONASS, and BEIDOU satellites, which reduces the time required to deliver distress alerts and positions and also allows better tracking of moving beacons [39].

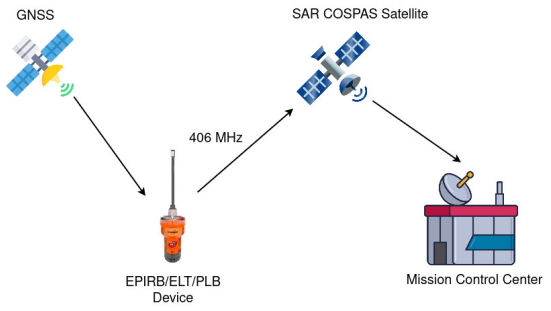


Figure 1: Very simplified COSPAS-SARSAT architecture

COSPAS-SARSAT uplink (Earth-to-space) operates on the 406–406.1 MHz frequency band and has a low baud rate of 400 bps, each message taking around 500ms to transmit [40]. COSPAS-SARSAT supports two types of messages: a short message of 112 bits and a long message of 144 bits, which are divided into various bit fields [40].

B. Elliptic Curve-based Digital Signature Algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a public key cryptography (PKC) algorithm that uses keys derived from an elliptic curve. ECDSA is based on the elliptic curve discrete logarithm problem (ECDLP). It has a short key and signature size. For example, an ECDSA 256-bit key signature provides equivalent security strength as an RSA 3072-bit key signature [41]. ECDSA operations at a very high level involving three main primitives, namely: key-pair generation, signature generation, and signature verification. Fundamental operations of these primitives are outlined in Appendix A.

C. Adversary and Threat Models

Below, we define the adversary capabilities and threat model of the proposed scheme and enumerate the security threats against the COSPAS-SARSAT 406 MHz protocols.

1) **Message Modification.** An adversary can use existing COSPAS-SARSAT 406 messages and modify their contents, resulting in *message-spoofing attacks* that transmit modified messages that appear authentic. Location spoofing can be an example of a message-modification attack in which the position data are intentionally altered. The absence of security measures poses a challenge for the COSPAS-SARSAT receivers (e.g., MCC, other nodes) in distinguishing between spoofed and genuine messages.
2) **Message Injection.** An adversary can falsify distress signals by replicating the COSPAS-SARSAT 406 message format and transmit them using inexpensive off-the-shelf transmitters (e.g., software-defined radios, or SDRs, such as HackRF), deceiving a legitimate receiver into accepting them as genuine. This would then cause fake distress signals to appear and give rise to what is known as a *ghost injection attack*. The reception and processing of these COSPAS-SARSAT 406 ghost distress signals may consume the resources of the MCC, and RCC, and influence whole rescue-chain decision-making.
3) **Replay Attacks.** In a replay attack, an adversary can intercept COSPAS-SARSAT 406 signals, record them, and subsequently retransmit the same messages and their corresponding signatures at a later time. This leads to various security risks, as it may result in the display of misleading and outdated information to the receiver (e.g., MCC).

In our model, the main goal of the attacker is to spoof legitimate (registered) targets by faking their identification and distress signals. The attacker also uses the *real actual GPS position* where the attacking SDR transmitter is located (as COSPAS-SARSAT uses radio frequency (RF) multilateration for signal-emitting signal positioning, thus can detect fake GPS position encoded in messages). The attacker aims to scale the attack in certain target areas (e.g., NATO borders) with cheap SDRs, static or on drones that can move at the attacker's will.

It is worth noting that the attacker capabilities described above are as realistic as possible, as demonstrated by numerous researchers [1], [3], [5], [42], [19], and there are several reasons for this. First, the protocols that we attempt to tackle are broadcast over unbound RF ranges. Thus, anyone has access to the communication. Second, the protocols are clear-text protocols (i.e., enable eavesdropping) and lack cryptographic means of message authentication and integrity (i.e., enable malicious injections and replays). Third, the low cost and accessibility of advanced SDRs enable great flexibility for their users, including malicious adversaries.

In addition to the threats mentioned above, other threats exist, such as eavesdropping, denial of service (DoS), and jamming attacks against COSPAS-SARSAT or MCC. Because we mainly focus on the authentication and integrity of the COSPAS-SARSAT data exchange, these other types of attacks are beyond the scope of this study, and protection against them needs to be studied in future work.

D. Security and Non-Security Goals of the Scheme

Considering the adversary capabilities defined above, our scheme aims to achieve the following **security goals**:

1) **Message Authentication.** Message authentication ensures that the incoming message is from an authenticated source and that the receiver can verify the source of the message. Our proposed scheme efficiently ensures the source authentication of the originating message. For example, a legitimate receiver (i.e., non-compromised via firmware attacks [43]) can verify whether the message received is from a legitimate source/transmitter.
2) **Message Integrity.** Message integrity means that the received data have not been modified while in transit, and any modification must be detected on the receiver side. In our scheme, the receiver detects any modifications in the received message. For example, a receiver can verify if a received message is tampered with or maliciously injected.
3) **Prevention of Replay Attack.** A replay attack is an attack in which an attacker $\mathscr{A}$ intercepts a message and maliciously retransmits or delays it. The proposed scheme prevents replay attacks by identifying and rejecting replayed messages based on an embedded timestamp.

Moreover, we set some **non-security goals** for our scheme:

1) **Backward Compatibility.** The main goal of the proposed scheme is to provide backward compatibility. First, this means that scheme implementation should be possible via **software-only upgrades** to the system, thus minimizing costs and disruption risks. Second, this means that the systems that were not upgraded would seamlessly discard the "New type" of messages and function "as expected" even in environments in which a mix of upgraded and non-upgraded nodes exists. We aim to achieve this by using "Spare Protocol" messages of COSPAS-SARSAT [40].
2) **Single Generic Approach.** Another goal of our scheme is to provide a single implementation that can be easily integrated into existing operational systems. For example, we envision our scheme as a single third-party software library plus minimal protocol-specific layers (e.g., extract node identity from protocol-specific messages). A single and minimal code-base is easier to audit and maintain, and can be applied to other domains (e.g., ADS-B).
3) **Minimum Communication Overhead.** Our scheme relies on sending digital signatures in follow-up messages, and we aim to keep the length of signed data as minimal as possible, which subsequently impacts the choice of digital signature algorithms and their corresponding key lengths. While increasing the key lengths is always an option in our scheme and implementation, we chose a minimal ECDSA key length in our current scheme that provides a sufficiently strong security guarantee (e.g., according to NIST [44]).

IV. Proposed Solution

In this section, we describe the proposed authentication and integrity scheme for the COSPAS-SARSAT 406 MHz. To simplify understanding, we use EPIRB devices and messages as an example. However, the scheme is generically applicable to all message types under COSPAS-SARSAT 406 MHz specifications [40]. The list of notations we have used in our proposed scheme is provided in Table I.

Table I: List of Notations

Symbol	Definition
G	Generator
k	Random number
R	Random point
$\mathscr{A}$	Adversary
$privKey$	Private key of the sender
$pubKey$	Public key of the sender
$epirb_m$	EPIRB type of COSPAS-SARSAT message
$cospassarsat_m$	COSPAS-SARSAT message
T_s	Timestamp from GPS/GNSS
ID	Explicit ID of the sender
h	Hash digest
σ	Signature

A. COSPAS-SARSAT Authentication

This section describes the source authentication and the integrity of COSPAS-SARSAT messages. The proposed scheme consist of two algorithms: "signature generation and encapsulation" and "signature verification and de-encapsulation". There are two types of COSPAS-SARSAT 406 Mhz digital messages: a short message type of 112-bit length and a

long message type of 144-bit length. Our proposed scheme for authentication and integrity of a 406 Mhz message uses an elliptic curve-based digital signature. In our scheme, to maintain the openness of the 406 MHz digital messages, the transmitter transmits signed messages in a follow-up "New type" of message. The transmitter emits normal messages and later sends the signature within follow-up long messages containing *Spare Protocol* payloads. The *Spare Protocol* is coded under "A2-B: Standard National, RLS and ELT(DT) Location Protocols" as either 0000 or 0001 [40]. The *Spare Protocol* contains the first 40 bits for synchronization, protocol, and country identification (Fig. 2).

We consider the rest of the 104-bit space to be the message field of the *Spare Protocol* to accommodate the signature and other information required by our scheme. The minimal signature size produced by the chosen ECDSA is 512 bits, and further concatenation of the timestamp of 64 bits and ID of 30 bits, making the total signature message 606 bits. In short, the timestamp is needed for replay detection and prevention (see Section IV-B), while the ID is used for node identity and private/public key pair authentication (see Section IV-B).

However, a 606-bit message cannot be accommodated in the 104-bit message field of the *Spare Protocol*. To overcome this message size limitation, we have added a feature in our *Spare Protocol* implementation to recognize and use a message-chaining mechanism.[3] Therefore, we propose and implement a chaining mechanism of 104 bits for the message field of the *Spare Protocol* to transmit a signed message in a follow-up message from the transmitter to the receiver.

Given that the signed message takes 606 bits, in practice, it requires six-chained COSPAS-SARSAT *Spare Protocol* messages to transmit the digital signature of one standard COSPAS-SARSAT message. COSPAS-SARSAT distress messages are generally event based, as they are sent only when the emergency occurs, which means that the 406 MHz RF channel is generally not congested (see also Section VI-A). Therefore, a 6x–7x message count increase (as required for a digital signature) should, most likely, not pose a challenge to the channel bandwidth and capacity while adding a strong "signal trustworthiness" factor. Thus, in our opinion, the tradeoff of packet increase (and temporary bandwidth impact) is well justified in this case.

1) Signature Generation/Encapsulation EPIRB Example:

The first algorithm in the proposed scheme is signature generation and encapsulation, as shown in Algorithm 1. An EPIRB message (either long or short) can be signed and encapsulated using this algorithm. First, it computes the GPS timestamp T_s, combine it with the EPIRB message, and then it computes the hash digest $h = hash(epirb_m||T_s)$ using the hash function $SHA-256$. The EPIRB device then computes the signature of the hashed value h using $privKey$ of the ECDSA cryptographic algorithm. To prevent a replay attack, the transmitter attaches the GPS timestamp T_s with signature σ. Then, it encapsulates a signed message using a proposed chaining mechanism with 104 bits into the *Spare Protocol* and transmits a "New type" of message that is supposed to be received and processed by legitimate receivers like MCC.

[3]Similar to the one available in the downlink format extended-length message (DF24 ELM) of ADS-B [45].

Algorithm 1 Signature Generation/Encapsulation – EPIRB example

procedure
Input: EPIRB message $epirb_m$, $SHA-256$
Output: Signature $\sigma = \{r, s\}$, Encapsulated EPIRB
Compute/Obtain GPS timestamp T_s
Calculate $h = hash(epirb_m||T_s)$ using $SHA-256$
Computes signature proof $\sigma = (privKey, h)$
Concatenate signature, timestamp, ID $\sigma \;||\; T_s \;||\; ID$
Chaining concatenated signed message into 104-bits of an available payload of EPIRB Spare Protocol
Transmits encapsulated EPIRB message to receivers (e.g., MCC)
end procedure

2) Signature Verification/De-encapsulation EPIRB Example:

When a legitimate receiver (e.g., MCC) ingests a signed message from the EPIRB source, it first de-encapsulates the *Spare Protocol* message received. It then obtains the message field containing the EPIRB-signed message concatenated with the timestamp and identification, as shown in Algorithm 2. The legitimate receiver combines the received timestamp T_s with the earlier received EPIRB message and computes the hash digest $h = hash(epirb_m||T_s)$ using SHA-256. Finally, the receiver validates the signature σ using $pubKey$. Validation of the signed message ensures that the received message is from an authenticated source and that the integrity of the message is preserved.

Algorithm 2 Signature Verification/De-encapsulation – EPIRB example

procedure
Input: Encapsulated EPIRB, $pubKey$, Message $epirb_m$, $SHA-256$
Output: Authenticated data
De-encapsulate EPIRB message and get Signed message $\sigma \;||\; T_s \;||\; ID$
Compute/Obtain GPS timestamp T_srecv and validate T_s is within T_srecv tollerable range
Calculate $h = hash(epirb_m||T_s)$ using $SHA-256$
Signature validation using public-key $(pubKey, h, \sigma)$
    if Signature is Verified **then**
        Keep received message: FULLY AUTHENTICATED
    else
        Possibly tampered/replayed, warn user: UNVERIFIABLE
    end if
end procedure

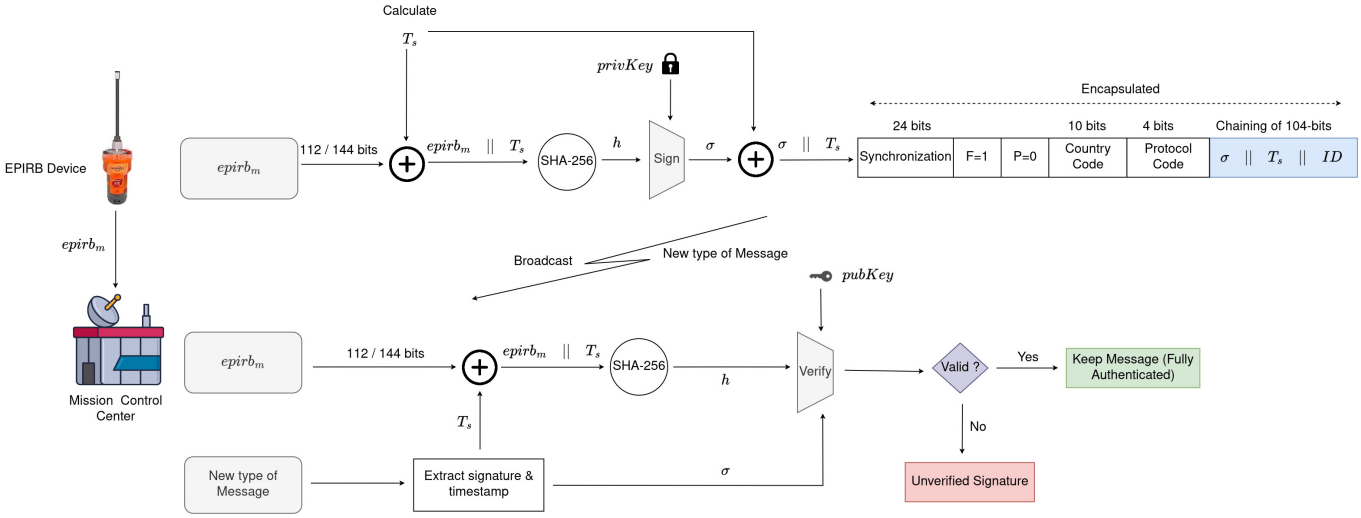


Figure 2: Our proposed COSPAS-SARSAT 406 MHz authenticity and integrity scheme – EPIRB example

B. Assumptions, Constraints, Recommendations

For clarity, we briefly outline and discuss some assumptions, limitations, and recommendations related to our scheme.

1) **Implicit Node Identity.** Our scheme assumes that the node identity (e.g., for public key loading, and other PKI requests) is built into protocol-specific identification field(s) of each addressed protocol, for example, the ID field within COSPAS-SARSAT.
2) **Explicit Node Identity.** Nevertheless, our scheme and implementation were designed to optionally allow, along with the digital signature and the timestamp, the inclusion of identification information, the ID field, into the "New type" of messages. This is a practical design choice for protocols in which some (or all) message types do not have an implicit node identification field. It is also practical to decouple the node identification from the public–private key pair, which may change several times during the node lifetime. For example, this ID field could be the prefix/suffix of the protocol's implicit node identity or the short key ID of the node's key pair.
3) **PKI Setup is Out of Scope.** Our scheme assumes that the PKI (e.g., key management) is already set up and readily available within the COSPAS-SARSAT ecosystem.
4) **Key Management is Out of Scope.** We assume that key management (i.e., generation, distribution, download, revocation, reissue, extension, expiration) is available to systems upgraded with our scheme. In other words, our scheme shall be able to call external services (e.g., PKI, key management) to at least request the public key of a node based on its domain-specific identity (e.g., ID field in COSPAS-SARSAT).
5) **Compromise of PKI or Private Keys is Out of Scope.** Our scheme assumes that the underlying PKI and private keys are safe and have not been compromised by attackers. Should any of the private keys be compromised, they should be revoked using standard PKI means and methods, and subsequent use of compromised private keys should be detected by all receivers, assuming the PKI is uncompromised and the use of PKI services is in line with the latest cybersecurity recommendations.
6) **Compromise of GPS and GNSS is Out of Scope.** In our scheme, we assume that GPS and GNSS are secure and continue to provide their services as expected. Cyber attacks capable of compromising these systems are out of the scope of this study.
7) **Jamming and Anti-jamming of SAR are Out of Scope.** In jamming, attackers create intentional low-level interference to block communication by legitimate nodes. Jamming/anti-jamming against COSPAS-SARSAT, MCC, and other nodes is out of the scope of this paper.
8) **Cryptographic Computations.** We assume that systems upgraded with our scheme have cryptographic computational capabilities (i.e., signature generation and verification) for beacon devices, the MCC, and other nodes of the system. Such computations can be facilitated through cryptographic modules (e.g., TPM), empowering beacon devices and MCC to execute cryptographic operations. Similar types of cryptographic modules for secure computations have been proposed for IoT [46], [47].
9) **Local Data Cache.** Our scheme also assumes that upgraded systems do not need to physically extend their local data storage. However, one aim is to recommend an upgraded system with additional (external) storage for signed messages, valid certificates, and public keys. While public keys and certificates can always be retrieved in real time, this would require internet connectivity to the PKI services and would introduce some time delay in the verification process. A local cache can solve these problems. The cache of the entire PKI is ideal but is less practical and more costly. Therefore, the cache can

contain only the public keys and certificates that a node is realistically expected to encounter during a route. As a future extension, we will explore integrating storage capability with the COSPAS-SARSAT system.

10) **Replay Prevention.** Our scheme assumes that a timestamp-based check (e.g., GPS timestamp) is sufficient to cover a good deal of replay attack attempts. While secure random "nonce"-based approaches (e.g., "nonce" challenge-response [48], predefined "nonce" sequences) provide stronger security guarantees and better replay protection, retrofitting strong-guarantee "nonce" support into existing protocols with backward compatibility is highly infeasible because of one-way broadcast COSPAS-SARSAT scenarios. Therefore, as a practical tradeoff, our scheme uses a timestamp-based check because the time of send is easy to append to the "New type" of messages carrying the signed message. For example, the use of GPS timestamps against replay attacks has been previously proposed in aviation ADS-B [49]. For further replay prevention improvements, see Section VI-C.
11) **Bit-wise Errors in Raw Messages.** Our scheme does not guarantee full resistance to bit-wise errors introduced through transmission–reception chains, the physical medium, or by malicious activity [3], [5], [42]. In theory, protocol-specific built-in error detection and recovery should be able to recover the sent messages, both the original message and the message containing the digital signature of the original message. We leave this nontrivial, comprehensive experimentation as a separate future work.
12) **Non-discard of Messages.** It is highly suggested that none of the messages, including those that fail signature verification, should be discarded, provided that this is balanced at runtime and the system has sufficient operational storage. These suggestions are based on the practical usability aspect of being able to later analyze the actual unverified messages in more detail to build a more robust system as well as devise improved attack countermeasures in the future. From a user interface (UI)/ user experience (UX) perspective: successfully verified messages could display a green checkmark, messages that fail the timestamp-based checks or fail to load the public key could display a yellow warning mark, and messages that fail signature verification (on successful public key load) could display a red error mark.

It is worth noting that some of the core assumptions are common and realistic according to the literature, whereas many security enhancement schemes exhibit similar limitations (e.g., not addressing PKI setup or key management in the scheme itself).

C. Security Modelling and Analysis

Below, we theoretically explore and evaluate the security strength of the proposed scheme based on our defined security goals and properties.

Theorem 1. *The legitimate receiver can detect false or tampered data values injected by external attackers.*

Proof. The transmitter computes the digital signature σ of its data using the private key $privKey$, encapsulates the signed message in the Spare Protocol, and sends it to the receiver. The receiver de-encapsulates the received signed message and validates the signature using $pubKey$ of the sender. Successful validation indicates that nobody tampered with the message. The scheme is secure against false or tampered data injection. □

Theorem 2. *Source authentication of incoming data to prevent ghost-injection attacks.*

Proof. An elliptic curve-based digital signature is used to authenticate the data source for each time slot. The sender computes a digital signature σ on its data using the private key $privKey$, attaches the current timestamp T_s, and transmits it to the legitimate receivers (e.g., MCC). Upon receiving a signed message, the legitimate receiver de-encapsulates and obtains the signed message from the message field, subsequently validating the signature by using the $pubKey$ of the sender. Successful validation of the signed message confirms that the received data originate from an authenticated source, thereby preventing the possibility of ghost-injected values by an attacker. The scheme prevents ghost injection attacks. □

Theorem 3. *Prevention of certain replay attacks.*

Proof. GPS timestamps (assuming GPS is not compromised) can be used as a form of forward-moving nonces [49]. Each sender computes the current GPS timestamp T_s, attaches it with the EPIRB message, and subsequently calculates the hash digest $h = \text{hash}(epirb_m||T_s)$ by using the SHA-256 hash function. The transmitter then generates signature σ, appends timestamp T_s and broadcast a message to the legitimate receivers (e.g., MCC). Upon message reception, the receiver extracts T_s and computes the hash digest $h = \text{hash}(epirb_m||T_s)$.

The receiver also computes/obtains its own GPS timestamp T_srecv to ensure and verify the "relative freshness" of the T_s (and its associated message therefore). If the difference is larger than a set threshold, the received T_s message could be flagged as stale and potentially replayed. This approach avoids a more complex challenge–response nonce exchange (which would not work easily in one-way broadcast COSPAS-SARSAT scenarios) but allows a very small window of opportunity for highly sophisticated attackers that are able to capture and replay a COSPAS-SARSAT message fast enough. Even if such an attack is successful, its impact would be limited since the duplication of messages will be limited and can also be filtered out using additional "anomaly detection" methods.

Finally, the receiver performs signature validation using the sender's public key. In the event of successful validation, the receiver retains the received message; otherwise, it discards it (or flags it as potentially replayed). Therefore, the proposed scheme effectively prevents certain replay attacks. □

V. Implementation Evaluation and Results

This section presents the evaluation results for the proposed COSPAS-SARSAT security enhancement scheme. We evaluated our scheme in terms of signature generation, signature verification, and overhead communication costs.

A. Computational Cost

We implemented our proposed scheme on a laptop operating Ubuntu 22 equipped with a 12th Gen Core i5 processor and 16GB of memory. The implementation leveraged Python-based *ecdsa* [50] and *hashlib* [51] cryptographic libraries. We conducted ten runs for each evaluation to obtain the average value in the experiment. The experiment involved measuring the time required to generate and verify *n* signatures, with *n* ranging from 1 to 1000. The signature generation cost for *n* signatures for COSPAS-SARSAT (i.e., for EPIRB samples) is depicted in Fig. 3a (Appendix B).

Similarly, we measured the time required to *verify* the *n* signatures for COSPAS-SARSAT (i.e., for EPIRB samples). The signing and verifying results show that the amount of time required increases with the number of signatures. The results of these experiments are presented in Fig. 3b (Appendix B).

B. Communication Cost

The proposed scheme's communication cost is determined by the number of bits transmitted in the authentication messages from the sender to the receiver. In our proposed scheme, the signed message has a 512-bit signature σ, 30-bit ID, and 64-bit timestamp T_s. This results in a signature message of 606 bits, requiring six-chained messages to transmit the signature of a single message. As illustrated in Fig. 3c (Appendix B), the communication overhead for the proposed scheme increases with the increasing number *n* of signatures. However, more evaluation and simulation would be required for multiple transmitters considering various probabilistic and message-collision models.

VI. Discussion

A. COSPAS-SARSAT 406 MHz Link Bandwidth and Capacity

At the end of 2021, COSPAS-SARSAT estimated the 406 MHZ beacon population to be between 2 and 3 million [52]. Moreover, "From September 1982 to December 2021, the Cospas-Sarsat System assisted in rescuing at least 57,413 persons in 17,663 SAR events" [52]. At the same time, a European Radiocommunications Committee (ERC) document from 1999 [53] reported 7,198 SAR alerts from 17 participants, with a staggering 6,773 of them (94,1%) classified as SAR false alerts. Due to missing descriptions of the methodologies in both documents, it is unclear how "SAR events" [52] relate to or are different from "SAR alerts" [53] and whether they relate to the actual EPIRB message count or not. *Thus, a possible future improvement would be to mandate a clear statistical methodology and the provision of statistics like open data, both globally and at country/center levels.*

On the one hand, currently, it may be safe to assume that the number of EPIRB messages/packets broadcast globally is within the range of several tens of thousands per year, which translates to a gross average of 30–100 messages/packets per day globally (i.e., for a single given geographic area the number of messages per day would be even lower). On the other hand, according to COSPAS-SARSAT C/S T.001 [40], [54], the messages are sent on the 406 MHz carrier at a baud rate of 400 bps, and the total transmission time (including unmodulated carrier) spans between 440–520 ms ($\pm 1\%$). This translates to a total available bandwidth of about 160,000 perfectly non-interfering messages for a 24-hour span within a single given geographic area covered by the beacon transmission power (e.g., beacons emit at 37dBm +/–2dBm (5W)). Given the limited 406 MHz bandwidth information so far, it is safe to assume that for the foreseeable future, the bandwidth of the 406 MHz channel is generally not congested, as opposed to ADS-B [55] and AIS [56], [57]. The uncongested nature of the channel is an important and enabling aspect when additional messages are required for the transmission of digital signatures authenticating previous normal transmissions.

> An important research aim to pursue in the immediate future is to independently explore and measure the usage, capacity, bandwidth, and congestion of the 406 MHz link, both from a global augmented view as well as from local regional perspectives.

B. Second Generation Beacons (SGB)

SGB leverages the MEOSAR space segment, enhancing detection probability, location accuracy, and system capacity. It employs a spread-spectrum modulation method that eliminates the need for channelization [58]. Each transmission burst has a longer message with more information, containing up to 202 message bits. SGB introduced the concept of *rotating message fields*, allowing different transmission bursts to carry different types of information. During a transmission burst, the beacon transmits one of 16 types of rotating message fields. Potentially in the future, 48-bit spare rotating fields can be used for the authentication of SAR messages by incorporating a signature or MAC inside these fields. We leave this as a future exploratory work.

C. Improved Anti-replay with Challenge-response Nonces

Galileo's return link service (RLS) and return link messages (RLMs) introduce two-way communication into COSPAS-SARSAT [59]. The intent of RLS/RLM is to provide the beacon and its user(s) a notification (e.g., blue LED light) that the COSPAS-SARSAT satellites and MCC received their message and are processing it, thus providing psychological assurance to the individuals affected by the distress situation. The RLS/RLM provide several "free-form" fields that do not have a strictly defined use.

RLS/RLM could potentially be used for two-way communication of challenge–response nonces for enhanced anti-replay protection. It could also help to optimize the usage of the limited RLS/RLM packet size to serve multiple challenge-response nonces–whether for multiple receivers or as a cache of nonces for future use for a single receiver. This is a promising research direction requiring further attention.

D. Jamming and Anti-jamming for COSPAS-SARSAT

Jamming and anti-jamming against COSPAS-SARSAT are beyond the scope of this study. The jamming resistance of COSPAS-SARSAT 406 MHz links could be improved naturally by the large-scale deployment of SGBs, where the spread-spectrum waveform is included in the specifications. Moreover, for researchers interested in investigating jamming/anti-jamming aspects of SAR, the ITU protection criteria for COSPAS-SARSAT 406 MHz can be a starting point, and we leave this research direction as a future work extension [60].

VII. Conclusion

We proposed a lightweight message authenticity and integrity scheme for **non-"secure by default"** COSPAS-SARSAT 406 MHz communications. The scheme is based on public/private key ECDSA state-of-the-art standards. The proposed scheme aims to retain the backward-compatibility and open nature of COSPAS-SARSAT by transmitting signed messages in follow-up "New type" of messages (piggy-backing on "Spare Protocol" availability in COSPAS-SARSAT) that non-upgraded systems can safely discard. However, further in vivo tests are required to confirm the backward compatibility properties. To the best of our knowledge, this work proposes and implements the (quasi-)first message authentication and integrity scheme for COSPAS-SARSAT 406 MHz communications. The lab experiments and results demonstrate the effectiveness of the proposed schemes in real-world scenarios. Moreover, we present a lightweight security analysis to demonstrate that our proposed scheme is secure under our threat model and can prevent the aforementioned types of attacks. Moreover, we identify and outline several research directions that are useful and interesting to pursue in the immediate future.

Acknowledgment

The authors thank all the NDSS SpaceSec24 anonymous reviewers and the technical program committee members for their valuable feedback, comments and information references that helped us improve the paper. In particular, the authors thank the paper's shepherd Dr. Martin Strohmeier for guiding and supporting this paper towards its best camera-ready shape.

Hannu Turtiainen thanks the Finnish Cultural Foundation / Suomen Kulttuurirahasto for supporting his Ph.D. dissertation work and research (grant decision no. 00231412).

The authors acknowledge the use of royalty-free icons in Figures 1, 2 courtesy of https://www.flaticon.com/ (icons by: DinosoftLabs, Wendy-G, juicy_fish, Freepik, Flat Icons, surang, Freepik, Peter Lakenbrink, Flat-icons-com, Handicon).

References

[1] Andrei Costin and Aurélien Francillon. Ghost in the air (traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices. *Black Hat USA*, 1, 2012.

[2] Martin Strohmeier, Matthias Schäfer, Vincent Lenders, and Ivan Martinovic. Realities and challenges of nextgen air traffic management: the case of ads-b. *IEEE Communications Magazine*, 2014.

[3] Syed Khandker, Hannu Turtiainen, Andrei Costin, and Timo Hämäläinen. Cybersecurity attacks on software logic and error handling within ads-b implementations: Systematic testing of resilience and countermeasures. *IEEE Transactions on Aerospace and Electronic Systems*, 2021.

[4] Marco Balduzzi, Alessandro Pasta, and Kyle Wilhoit. A security evaluation of ais automated identification system. In *Proceedings of the 30th annual computer security applications conference*, 2014.

[5] Syed Khandker, Hannu Turtiainen, Andrei Costin, and Timo Hämäläinen. Cybersecurity attacks on software logic and error handling within ais implementations: A systematic testing of resilience. *IEEE Access*, 2022.

[6] D. L. M. Dsc and D. P. Lewis. Space, the final frontier for cybersecurity? https://www.chathamhouse.org/2016/09/space-final-frontier-cybersecurity. (Visited on 03-01-2024).

[7] James Pavur and Ivan Martinovic. Sok: Building a launchpad for impactful satellite cyber-security research. *arXiv preprint arXiv:2010.10872*, 2020.

[8] G. Falco. No access the vacuum of space cyber security. https://arc.aiaa.org/doi/10.2514/6.2018-5275. (Visited on 03-01-2024).

[9] Lishoy Francis, William G. Sirett, Keith Mayes, and Konstantinos Markantonakis. Countermeasures for attacks on satellite tv cards using open receivers. In *Proceedings of the 2005 Australasian Workshop on Grid Computing and E-Research - Volume 44*, ACSW Frontiers '05. Australian Computer Society, Inc., 2005.

[10] M.P. Howarth, S. Iyengar, Z. Sun, and H. Cruickshank. Dynamics of key management in secure satellite multicast. *IEEE Journal on Selected Areas in Communications*, 2004.

[11] Yingli Sheng, Haitham Cruickshank, Martin Moseley, and John Ashworth. Security architecture for satellite services over cryptographically heterogeneous networks. In *2011 6th International ICST Conference on Communications and Networking in China (CHINACOM)*, 2011.

[12] R.J. Hughes, W.T. Buttler, P.G. Kwiat, S.K. Lamoreuax, G.L. Morgan, J.E. Nordholt, and C.G. Peterson. Quantum cryptography for secure satellite communications. In *IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, 2000.

[13] Maqsood Ahmed. Satellite-aided search and rescue (sar) system. *IEEE Aerospace and Electronic Systems Magazine*, 2007.

[14] D Levesque. The cospas-sarsat system. In *IEE Colloquium on Satellite Distress and Safety Systems*. IET, 1993.

[15] COSPAS-SARSAT.INT. Cospas-sarsat international satellite system for search and rescue. https://www.cospas-sarsat.int/en/.

[16] Andrea A Serra, Paolo Nepa, and Giuliano Manara. A wearable two-antenna system on a life jacket for cospas-sarsat personal locator beacons. *IEEE Transactions on antennas and propagation*, 2011.

[17] GARY Vrckovnik and CR Carter. 406 mhz elt signal spectra for sarsat. *IEEE transactions on aerospace and electronic systems*, 1991.

[18] BoatU.S. Foundation. How epirbs work. https://www.boatus.org/epirb/work/.

[19] Andrei Costin, Syed Khandker, Hannu Turtiainen, and Timo Hämäläinen. Cybersecurity of COSPAS-SARSAT and EPIRB: threat and attacker models, exploits, future research. In *Workshop on Security of Space and Satellite Systems (SpaceSec) 2023, NDSS*, 2023.

[20] Dejan V Kožović, Dragan Ž Đurđević, Mirko R Dinulović, Saša Milić, and Boško P Rašuo. Air traffic modernization and control: Ads-b system implementation update 2022: A review. *FME Transactions*, 2023.

[21] Darren Wright, Carol Janzen, Robert Bochenek, Jessica Austin, and Edward Page. Marine observing applications using ais: Automatic identification system. *Frontiers in Marine Science*, 2019.

[22] Syed Khandker, Krzysztof Jurczok, and Christina Pöpper. (TO APPEAR) COSPAS Search and Rescue Satellite Uplink: A MAC-Based Security Enhancement. In *Workshop on Security of Space and Satellite Systems (SpaceSec) 2024, NDSS*, 2024.

[23] Pingyue Yue, Jianping An, Jiankang Zhang, Jia Ye, Gaofeng Pan, Shuai Wang, Pei Xiao, and Lajos Hanzo. Low earth orbit satellite security and

reliability: Issues, solutions, and the road ahead. *IEEE Communications Surveys & Tutorials*, 2023.

[24] Yuqi Lv, Qifeng Ding, Xiaoyong Liu, Junchi Zhang, and Hua Yang. Interference analysis of the public walkie-talkie on the cospas-sarsat system's uplink. In *2020 15th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 2020.

[25] Jacob Krabbe Pedersen, Mikkel Bøchman, and Weizhi Meng. Security analysis in satellite communication based on geostationary orbit. In *19th Annual International Conference on Privacy, Security & Trust (PST)*. IEEE, 2022.

[26] National Institute of Standard and Technology (NIST). Nist sp 800-30 rev. 1: Guide for conducting risk assessments. https://csrc.nist.gov/pubs/sp/800/30/r1/final.

[27] Chunxiao Jiang, Xuexia Wang, Jian Wang, Hsiao-Hwa Chen, and Yong Ren. Security in space information networks. *IEEE Communications Magazine*, 2015.

[28] Mark Manulis, Christopher P Bridges, Richard Harrison, Venkkatesh Sekar, and Andy Davis. Cyber security in new space: analysis of threats, key enabling technologies and challenges. *International Journal of Information Security*, 2021.

[29] James Pavur. *Securing new space: on satellite cyber-security*. PhD thesis, University of Oxford, 2021.

[30] Stavros Stavrinos, Konstantinos Kotis, and Christos Kalloniatis. Towards semantic modeling and simulation of cybersecurity on the internet of underwater things. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2022.

[31] Ulrike Lechner. It-security in critical infrastructures experiences, results and research directions. In *Distributed Computing and Internet Technology: 15th International Conference, ICDCIT*. Springer, 2019.

[32] Ignacio Martinez-Alpiste, Gelayol Golcarenarenji, Qi Wang, and Jose Maria Alcaraz-Calero. Search and rescue operation using uavs: A case study. *Expert Systems with Applications*, 2021.

[33] Karin Bernsmed, Guillaume Bour, Per Håkon Meland, Ravishankar Bhaskarrao Borgaonkar, and Egil Wille. D4. 3 multi-modal communication-securing future communication across different sectors and technologies. *SINTEF AS (ISBN starter med 978-82-14-)*, 2021.

[34] Inone Joo, Jeom-Hun Lee, Young-Min Lee, Cheon Sig Sin, Sang-Uk Lee, and Jae Hoon Kim. Development and performance analysis of the second generation 406 mhz epirb. In *4th Advanced Satellite Mobile Systems*. IEEE, 2008.

[35] JV King. Cospas-sarsat: an international satellite system for search and rescue. *Space communications*, 2002.

[36] COSPAS-SARSAT.INT. Description of the 406-mhz payloads used in the cospas-sarsat leosar system, c/s t.003 issue 5 – revision 1 march 2022. https://www.cospas-sarsat.int/images/stories/SystemDocs/Current/T003-MAR-25-2022.pdf.

[37] COSPAS-SARSAT.INT. Description of the 406-mhz payloads used in the cospas-sarsat geosar system, c/s t.011 issue 2 – revision 4 october 2023. https://www.cospas-sarsat.int/images/stories/SystemDocs/Current/T011-OCT-27-2023.pdf.

[38] COSPAS-SARSAT.INT. Description of the 406 mhz payloads used in the cospas-sarsat meosar system, c/s t.016 issue 1 - revision 7 october 2023. https://www.cospas-sarsat.int/images/stories/SystemDocs/Current/T016-OCT-27-2023.pdf.

[39] Dany St-Pierre-Cospas-Sarsat Secretariat. Beacon manufacturers workshop jacksonville, florida, 16 june 2023. https://www.sarsat.noaa.gov/wp-content/uploads/2023-BMW_Cospas-Sarsat_Update.pdf. (Visited on 7-03-2024).

[40] COSPAS-SARSAT.INT. Specification for cospas-sarsat 406 mhz distress beacons – c/s t.001, Mar 2022.

[41] Svetlin Nakov. *Practical Cryptography for Developers*. SoftUni, November 2018.

[42] Syed Khandker, Hannu Turtiainen, Andrei Costin, and Timo Hämäläinen. On the (In) Security of 1090ES and UAT978 Mobile Cockpit Information Systems–An Attacker Perspective on the Availability of ADS-B Safety-and Mission-Critical Systems. *IEEE Access*, 2022.

[43] Andrei Costin, Jonas Zaddach, Aurélien Francillon, and Davide Balzarotti. A {Large-scale} analysis of the security of embedded firmwares. In *23rd USENIX security symposium (USENIX Security 14)*, 2014.

[44] Hildegard Ferraiolo and Andrew Regenscheid. Cryptographic algorithms and key sizes for personal identity verification. Technical report, National Institute of Standards and Technology, 2023.

[45] Junzi Sun. The 1090 megahertz riddle: a guide to decoding mode s and ads-b signals. *TU Delft OPEN Publishing*, 2021.

[46] Ke Han, Youyan Duan, Rui Jin, Wendou Wu, Baijuan Wang, and Xiaobo Cai. Optimal design of encryption module in iot. *Procedia Computer Science*, 2021.

[47] Stefano Di Matteo, Luca Baldanzi, Luca Crocetti, Pietro Nannipieri, Luca Fanucci, and Sergio Saponara. Secure elliptic curve crypto-processor for real-time iot applications. *Energies*, 2021.

[48] Geir M Køien. A brief survey of nonces and nonce usage. In *SECURWARE international conference on emerging security information, systems and technologies*, 2015.

[49] Wei-Jun Pan, Zi-Liang Feng, and Yang Wang. Ads-b data authentication based on ecc and x. 509 certificate. *Journal of Electronic Science and Technology*, 2012.

[50] Pure-python ecdsa and ecdh. https://pypi.org/project/ecdsa/.

[51] Python docs: Secure hashes and message digests. https://docs.python.org/3/library/hashlib.html.

[52] COSPAS-SARSAT.INT. Cospas-sarsat system data no.48 - december 2022. https://cospas-sarsat.int/images/stories/SystemDocs/Current/SD48-DEC22--EN-.pdf.

[53] European Radiocommunications Committee (ERC). Handling and usage of emergency position indicating radio beacon (epirb) to prevent false alerts. https://docdb.cept.org/download/2224.

[54] William D Ivancic. Cospas/sarsat 406-mhz emergency beacon digital controller. Technical report, NASA, 1988.

[55] T Verbraak, Joost Ellerbroek, Junzi Sun, and Jacco Hoekstra. Large-scale ads-b data and signal quality analysis. In *Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar*, 2017.

[56] Savio Sciancalepore, Pietro Tedeschi, Ahmed Aziz, and Roberto Di Pietro. Auth-ais: secure, flexible, and backward-compatible authentication of vessels ais broadcasts. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[57] Menghui Yang, Yongzhong Zou, and Li Fang. Collision and detection performance with three overlap signal collisions in space-based ais reception. In *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2012.

[58] COSPAS-SARSAT.INT. Specification for second-generation cospas-sarsat 406-mhz distress beacons, c/s t.018 issue 1 – revision 11 november 2023. https://cospas-sarsat.int/images/stories/SystemDocs/Current/T018-OCT-27-2023.pdf.

[59] EUROPEAN GNSS (GALILEO). Sar/galileo service definition document. https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo-SAR-SDD.pdf.

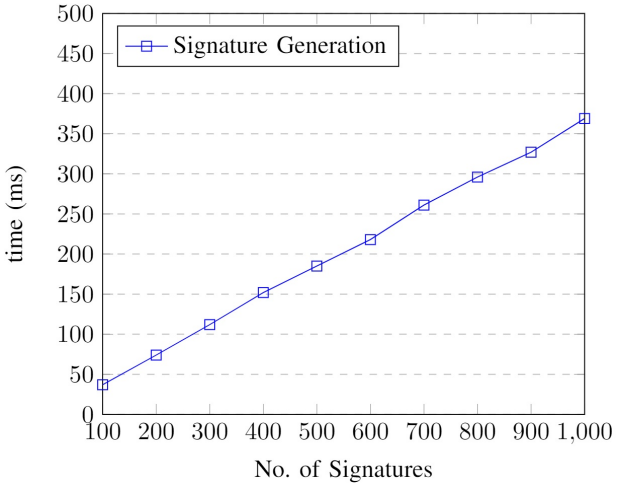
[60] ITU. Protection criteria for cospas-sarsat search and rescue instruments in the band 406-406.1 mhz-rec. itu-r m.1478-1. https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1478-1-200405-S!!PDF-E.pdf.

APPENDIX

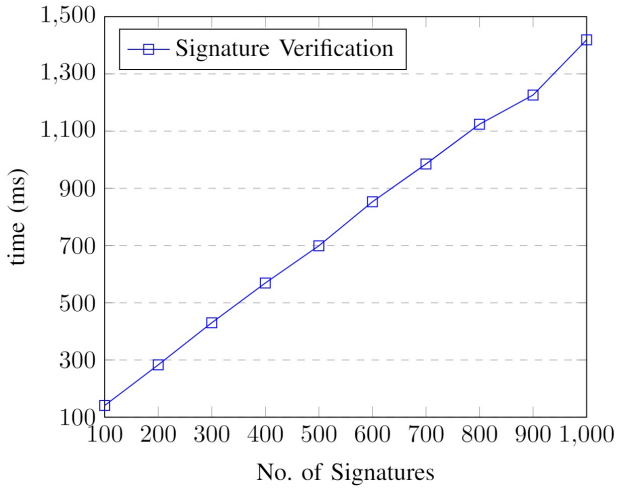
A. Overview of ECDSA Fundamentals

- **Key-Pair Generation:** Securely generate a Private Key as a secure-random integer in the $[0....n-1]$ range. Then calculate Public Key as $pubKey = privKey \times G$.
- **Signature Generation:** Calculate hash of any message m using a secure hash of the message (e.g., $SHA-256$) as $h = hash\,(m)$. Then generate a random number k in the range $[1...n-1]$, calculate random point $R = k \times G$, and take its x-coordinate $r = R\,.\,x$. Computes signature proof $s = k^{-1} \times (h + r \times privKey)\,(mod\,n)$ and returns signature $\sigma = \{r, s\}$.
- **Signature Verification:** Calculate hash of message m, as $h = hash\,(m)$ using same secure hash algorithm (e.g., $SHA-256$). Then calculate modular inverse $s1 = s^{-1}\,(mod\,n)$ and recover random point $R^{'} = (h \times s1) \times G + (r \times s1) \times pubKey$ and take its x-coordinate $r^{'} = R^{'}\,.\,x$ and then finally perform signature validation by comparing $r^{'} == r$.

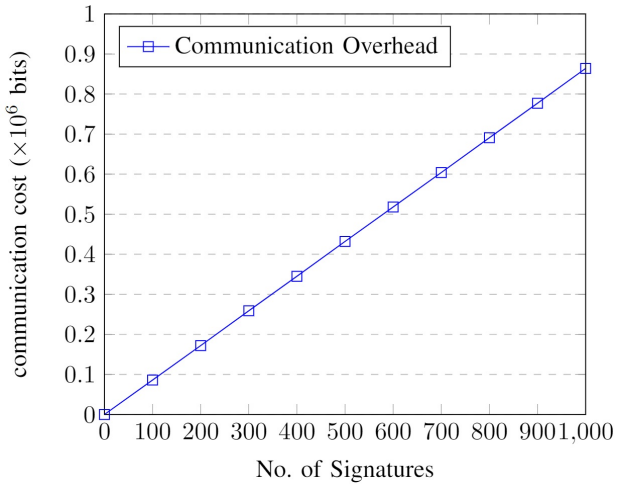
B. Evaluation Results



(a) Signature Generation Cost



(b) Signature Verification Cost



(c) Communication Cost

Figure 3: Evaluation Results