Inherent Vulnerabilities in Hybrid CDMA & Cryptographic Spread Spectrum for Space Systems

Edd Salkield
University of Oxford
edd.salkield@cs.ox.ac.uk

Sebastian Köhler
University of Oxford
sebastian.kohler@cs.ox.ac.uk

Simon Birnbach
University of Oxford
simon.birnbach@cs.ox.ac.uk

Ivan Martinovic
University of Oxford
ivan.martinovic@cs.ox.ac.uk

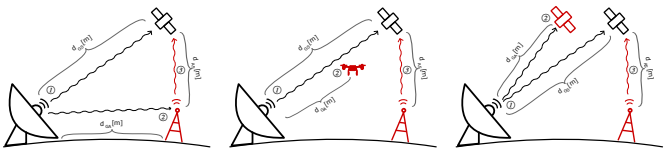
***Abstract*—Direct Sequence Spread Spectrum (DSSS) is used to simplify frequency management for constellations and for use of data relay satellites, to improve satellite mission availability against unintentional interference and protect space RF links against jamming, eavesdropping, and spoofing. Whilst current standards focus on cooperative Code Division Multiple Access (CDMA) DSSS methods, high-value government and military assets increasingly use cryptographic DSSS to improve security. Including cryptographic DSSS into future revisions of the ETSI standard is currently considered an option, but it has been found that cryptographic DSSS is significantly worse at multiple access than the currently standardized methods. In this context, the European Space Agency and Thales Alenia Space have studied a *hybrid* CDMA/cryptographic DSSS construction designed to simultaneously provide multiple-access and security.**

In this paper we perform the first systematic analysis of the hybrid protocol and discover a number of major design flaws which are fundamental to the design and seriously degrade the security of the system. In particular, we find that reuse of the cryptographic spreading sequence leads to a catastrophic failure wherein all satellites' data sequences can be recovered with high probability given knowledge of any single satellite's data sequence. This also enables sufficient recovery of the spreading sequence to spoof arbitrary messages, and increases vulnerability to optimized jamming. We evaluate and validate these findings through simulations with respect to real-world systems, and use this to propose countermeasures and system improvements which should be considered as standardization work continues.

***Index Terms*—spread spectrum, security analysis, physical-layer security.**

I. Motivation

Direct Sequence Spread Spectrum (DSSS) is used and standardized for protecting the uplink, downlink and ranging transmissions of satellite communications [1, 2]. By assigning each satellite a unique spreading sequence, each satellite can correlate just for its own sequence thereby reducing accidental interference between concurrent data streams. This system is known as Code Division Multiple Access (CDMA) and relies on the cooperation of different space agencies in assigning sequences as coordinated through SFCG [2, 3]. This system is critical in decreasing interference in shared ground station and relay network settings such as NASA's TDRSS, and reducing the power spectral density to remain within license restrictions [2]. However, whilst such a system may appear to also provide security properties at the physical layer[1], including unobservability, secrecy, authenticity, and anti-jamming availability, many studies have shown that these are easily defeated by adversaries which know the spreading sequence or can derive it through a number of available "blind estimation" methods [4–8].



(a) Ground-based (b) In-beam (c) Satellite

Fig. 1. The three adversary scenarios considered in this work. Since advanced attacks against secrecy, authenticity, and availability rely on an eavesdropping component, the key distinction is the placement of the eavesdropping receiver with respect to the ground station.

Cryptographic DSSS seeks to address this issue by using a secure stream cipher to generate a random spreading sequence (often referred to as a pseudo-noise or PN sequence), thereby preventing the attacker from deriving it [9]. Currently cryptographic DSSS is used and is available off-the-shelf for military satellites [10]. ETSI has indicated that the study of cryptographic DSSS for civil missions may be resumed for future standard revisions[2] [11]. However, it has been noted that cryptographic DSSS provides significantly worse multiple access properties than the linear CDMA schemes standardized by ETSI and in current widespread non-military use, in certain cases up to 30 dB of additional interference [12].

To address these limitations, the European Space Agency (ESA) commissioned a study in which a hybrid CDMA/cryptographic construction was proposed by Thales Alenia Space - Italy (TAS-I) [12, 13]. The hybrid scheme is intended to provide both the multiplexing properties of CDMA and the security properties of cryptographic DSSS. The Final Report for this work was completed in 2011 and was subsequently updated in 2021 for a later round of ESA project bidding [12, 14]. Tests were performed using TAS-I's Spread Spectrum Transponder architecture which provide "fully customizable TRANSEC codes and spreading rates" [15]. Responsible disclosure to TAS-I has revealed that

[1]In the literature, physical layer security is often referred to as TRANSEC, or Transmission Security.

[2]Current ETSI standards for Satellite Earth Stations and Systems (SES) consider the use and allocation of spreading code sequences for achieving multiple access and anti-jamming in TT&C scenarios.

this scheme is not currently implemented in their products. However the need for protocols which support security and multiple access is growing as the number of deployed satellites is rapidly increasing, as evidenced by recent hybrid protocol proposals [16]. Therefore a comprehensive security analysis is highly desirable.

In this paper, we make the following contributions:

- We systematize and assess the security properties of the hybrid DSSS scheme studied by ESA.
- We identify inherent vulnerabilities in the scheme which break the system's unobservability, secrecy, and authenticity, and degrade its anti-jamming availability.
- We propose novel methods by which these attacks can be conducted by low-capability adversaries, and evaluate the effectiveness of these approaches with respect to the real-world systems shown in Figure 1.
- We use our results to propose new countermeasures to increase the security of the system.

II. Related work

The hybrid construction studied by ESA is the first known combination of a cryptographic spreading sequence with maximum length sequences, but the principle of combining a cryptographic sequence with CDMA has been explored in other context. This system construction must not be confused with other hybrid spread spectrum systems which combine Direct Sequence with Frequency Hopping [17, 18].

Cryptographic/DSSS is the principle behind secure-mode GNSS systems such as GPS, where it was found that additional information can be extracted from the signal without knowledge of the key [19]. It is known that secure-mode GNSS, alongside DSSS in general, is vulnerable to replay attacks even if the attacker cannot detect the signal [20, 21]. One recent study proposed a mixed AES/Gold Sequence approach to TRANSEC, but its security analysis considers jamming only and ignores other physical-layer security properties [16].

A number of works consider attacks against non-hybrid DSSS which apply equally to the hybrid context. For instance, for long non-cryptographic chip sequences a reactive jamming strategy can be employed in which the attacker eavesdrops on the first part of the spreading sequence to decide whether or not to jam the bit in the latter part [22]. We consider this technique in Section V-D. Repeater jammers have also been considered in which a jammer receives and immediately retransmits the same signal with inverted phase to effect signal cancellation and/or introduce errors [23, 24]. However, this technique is challenging for space systems since long path lengths introduce high latency [14].

Outside of spread spectrum, other physical-layer security measures such as adaptive coding and modulation have been shown to be exploitable, weakening overall security [25].

III. System Model

The hybrid spread spectrum system studied by ESA is designed to allow a single ground station to communicate securely and with good multiplexing properties with several satellites simultaneously. The security properties are provided at the Physical/Transmission Layer, and are known as TRANSEC protections. The hybrid system is distinct from a true multiple access scheme which would support multiple simultaneous ground stations. We first explain the construction of the system, and then formalize its security objectives.

A. Hybrid spread spectrum system

Each satellite is assigned a unique ML value of length N from a *maximum-length sequence set*, a collection of pseudo-random binary sequences, which acts as a unique identifier. To construct this set, a single size N maximum-length sequence ML is generated and rotated under every possible bit-rotation. This is modulated into BPSK symbol space where each 1 bit is represented by symbol -1, and each 0 bit is represented by symbol 1[3]. Going forward, we use $\tilde{x}$ when in symbol space and x when in bit space.

The key property that makes maximum-length sequences useful for multiplexing is that the correlation (dot product) between an $\tilde{ML}$ in the set and itself takes value $\tilde{ML}_i \cdot \tilde{ML}_i = N$, whereas $\tilde{ML}_i \cdot \tilde{ML}_j = -1$ where $i \neq j$. We use $\tilde{ML}_i$ to denote the rotation assigned to the i^{th} satellite.

A communication session proceeds in three stages as illustrated in Figure 2. First, a well-known *Long Acquisition Code* (LAC) is transmitted, allowing all satellites to synchronize to and sample the chips within the signal. Second, the *Synchronization Data* (SD) is transmitted acting as the session key. For each satellite i, the same cryptographic pseudo-noise sequence $\tilde{PN}$ is generated using a secure stream cipher with the shared key k and SD. The per-satellite pseudorandom spreading sequence is the multiplication (XOR) of $\tilde{PN}$ and $\tilde{ML}_i$.

$$\tilde{PN}_i = \tilde{PN} \times \tilde{ML}_i \tag{1}$$

Finally, the protected *Secure Communication* can begin. The basic idea is that, given a constant spreading factor M, a 0 bit is communicated by transmitting M BPSK symbols (known as "chips") of PN_i and a 1 bit by the inverse of PN_i. More formally, each bit is repeated over M consecutive chips to form D_i, and the transmitted chips are formed by taking the multiplication (Hadamard product) $D_i \times PN_i$. Each satellite i recovers its sequence by correlating for $\tilde{PN}_i$ and measuring the sign of the result. In the absence of data, the Final Report specifies that an idle zero sequence is sent.

The ground station then transmits the chip sequence for each satellite simultaneously and synchronously. Depending on the required SNR at each specific satellite, the gain g_i of the i^{th} satellite's chip sequence can be independently adjusted.

Thus in symbol space the final signal is the composite (sum) of the chip sequences for all satellites, each of which consists of the multiplication of their components plus a Gaussian channel noise component with noise power N_0:

$$\tilde{sig} = g_1 \cdot \tilde{PN}_1 \times \tilde{D}_1 + \cdots + g_n \cdot \tilde{PN}_n \times \tilde{D}_n + n \quad \text{where} \quad n \sim \mathcal{N}(0, N_0) \tag{2}$$

[3]This mapping is chosen so that an XOR in bit space is equivalent to multiplication (Hadamard product) in symbol space.

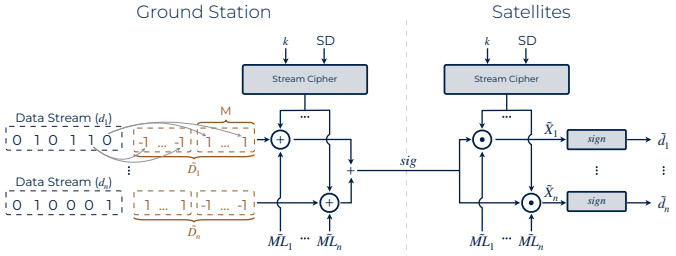


Fig. 2. Overview of the TRANSEC mechanism. The transmitting ground station architecture is illustrated on the left, and the receiving satellite architecture on the right. The key weakness of the system stems from reuse of the PN sequence, generated from the Stream Cipher, for all satellite data streams.

Each receiver correlates $\tilde{sig}$ with its spreading sequence $\tilde{PN}_i$ in order to recover the data component $\tilde{X}_i$. The sign of $\tilde{X}_i$ indicates the bit that was sent. An illustration of this signal and the transmit-receive process is provided in Figure 2.

B. Security properties

This system is designed to provide security properties at the physical layer alongside the multiplexing properties of CDMA. Being a TRANSEC security mechanism, the primary security property is availability against jamming which *communication security* (COMSEC) methods at higher protocol layers cannot implement. In addition, this mechanism is concerned with providing unobservability, a form of secrecy against eavesdropping which prevents analysis of the traffic flows even when encryption is used at higher layers. Finally, authenticity against spoofing is an additional proposed benefit, although this is typically implemented at higher protocol layers.

The hybrid mechanism is intended to provide anti-jamming availability based on the principle that without knowledge of the pseudo-noise spreading code, all jamming signals can only be as effective as a bandwidth-matched Gaussian jammer [26]. The intuition given for the secrecy of this system is that each data component has been XORed with a true pseudorandom sequence PN which cannot be predicted by the attacker[4]. Since each communication session begins with a well-known LAC, unobservability is not provided, and the system is immediately opened up to protocol-aware jamming [27]. The system is considered secure against spoofing if the attacker is unable to generate authentic messages without knowledge of the secret key; we note that authenticity may still be provided by higher layer COMSEC methods.

IV. Threat Model

We consider an attacker with the goal of breaking the security properties of the system: namely unobservability and secrecy via eavesdropping, authenticity via spoofing, and availability via jamming. We assume that the attacker has prior knowledge of fixed protocol parameters including the publicly-known LAC, format of the SD, and spreading factor M. Notably, we do not assume that the attacker has knowledge of the secret key k. Therefore they are unable to generate the cryptographic pseudo-noise sequence PN. We consider both cases where the attacker has no prior knowledge of any data sequence and where the attacker knows a single satellite's data sequence. We note that well-known sequences of are selected for idle periods: all-zero in the Final Report, and alternating zeros and ones in the CCSDS standards [12, 28].

Previous works have also shown that fixed parts of the protocol such as known headers can be exploited [27]. At the physical layer we assume that the attacker has suitable equipment including a sufficiently wideband software-defined radio to match the transponder. We consider spoofing via overshadowing, where no synchronization to an existing data stream is required [29]. When jamming we assume that the attacker can synchronize to the data bits but not the individual chips, allowing reactive jamming to occur. Bit synchronization is easier than chip synchronization as the spreading factor is typically very high, with $M \approx 2^{22}$ chips, and is a commonly considered threat model [22, 30]. We consider different levels of proximity between the attacker and ground station, which impacts the SNR at which the attacker can receive the signal.

V. Attack

The hybrid protocol is inherently vulnerable to eavesdropping and spoofing attacks, despite all data sequences being encrypted with a pseudorandom sequence PN. In contrast to cryptographic DSSS, it is also susceptible to optimized jamming. These attacks are possible without breaking any cryptographic operation and without knowledge of the key.

The fundamental vulnerability is caused by the reuse of the same PN for all data sequences, meaning each satellite's $\tilde{PN}_i$ consists of a per-satellite maximum-length spreading sequence $\tilde{ML}_i$ combined with the same PN (see Equation 1). This allows the attacker to extract mutual information about the content of any data bits transmitted simultaneously. In particular, knowledge of a single satellite's data sequence is sufficient to recover the data sequence of all other satellites in the system with high probability. Furthermore, since the waveform contains repeated aggregate chips of the same magnitude, the attacker does not require a high gain antenna to recover the chips. This is a catastrophic outcome since the synchronization data (SD) does not provide freshness guarantees and can therefore additionally be reused by the attacker for spoofing.

We proceed to describe the attack in three stages. First, we demonstrate that the attacker can receive an aggregate of the chip sequences without prior knowledge either of PN or the satellite ML sequences. Instead only the knowledge that the ML sequence repeats is required. Second, we show that the relationship between all bits transmitted simultaneously can be inferred from the aggregate chips, and reduces in most cases to two bit combinations such that knowledge of any one bit reveals all other simultaneous bits. We then show that the attacker can derive an estimate of PN which is sufficient to

[4]The Report states "the hybrid spreading sequence has the same cryptographic strength as the cryptographically strong PN sequence... This follows from the fact that the Hadamard product of a ±1 coin-tossing sequence with any ±1 sequence is another ±1 coin-tossing sequence."

allow the encoding of attacker-controlled messages. Finally we propose a new method for using the mutual information to optimize a jammer waveform against the system.

A. Receiving aggregate chips

As the signal $\tilde{sig}$ is encoded with a spreading code, we assume that the signal is beneath the noise floor such that the data is too noisy to reliably measure the state of any individual chip. A receiver with knowledge of the cryptographic spreading sequence $\tilde{PN}_i$ overcomes this by measuring the state of the data sequence over a long-run average.

Recall the signal from Equation 2: the multiplication (XOR in bit space) of each data sequence $\tilde{D}_i$ (for satellite i) with the per-satellite pseudo-noise sequence $\tilde{PN}_i$, where there are m satellites in total and the noise at a given receiver is n. We can rewrite this, factoring out the common $\tilde{PN}$ sequence and grouping together the satellite-specific components:

$$\begin{aligned}\tilde{S} &= g_1 \cdot \tilde{PN} \times \tilde{ML}_1 \cdot \tilde{D}_1 + ... + g_m \cdot \tilde{PN} \cdot \tilde{ML}_m \times \tilde{D_m} + n \\ &= \tilde{PN} \times \left(\tilde{g_1} \cdot \tilde{C_1} + ... + \tilde{g_m} \cdot \tilde{C_m} \right) + n = \tilde{PN} \times C + n \quad (3) \\ &\text{where } \tilde{C_i} = \tilde{ML}_i \times \tilde{D}_i \text{ and } \tilde{C} = \tilde{g_1} \cdot \tilde{C_1} + ... + \tilde{g_m} \cdot \tilde{C_m}\end{aligned}$$

Since $\tilde{PN}$ is a pseudorandom sequence of 1 and -1, its overall effect on each data chip in the signal is to randomize its sign. Whilst this is sufficient to encrypt a data sequence from a single satellite, information is leaked under multiple sequences because the sign change affects all simultaneous data chips in the same way. Therefore the overall effect is to randomize the sign of the aggregate chip $\tilde{C}$ but not its contents.

Since $\tilde{PN}$ is constructed of repeating per-satellite sequences $\tilde{PN}_i$, we can estimate the magnitude of the chips in $\tilde{C}$ over a long-run average by noting that every aggregate chip magnitude in $\tilde{C}$ is repeated M/N times, where M is the spreading factor of the pseudo-noise PN and N is the length of the per-satellite ML sequences. Specifically:

$$\begin{aligned}&\tilde{C_i}(t) = \tilde{C_i}(t') \quad \text{if} \quad t' \in \text{repeats}(t) \quad (4) \\ &\quad \text{where} \quad \text{repeats}(t) = \{t' \mid t\%N = t'\%N \text{ and } t/M = t'/M\} \\ &\quad \text{and} \quad \#\text{repeats}(t) = M/N\end{aligned}$$

The estimate $\tilde{C}_{est}$ is given by accumulating all repeated chips into a buffer and taking the long-run average of its absolute values. This decreases the effective noise power noise in proportion to the number of repeats, M/N:

$$\begin{aligned}\tilde{C}_{est}(t) &= \pm \frac{1}{\#\text{repeats}(t)} \cdot \sum_{t' \in \text{repeats}(t)} |\tilde{sig}(t')| \quad (5) \\ &= \pm \tilde{C}(t) + \mathcal{N}\left(0, \frac{N}{M} \cdot N_0\right) \quad (6)\end{aligned}$$

B. Recovering the data sequences

Given an estimate of the aggregate chips $\tilde{C}_{est}$, we can deduce the most likely data bits that could have been transmitted. This can be done through Maximum Likelihood decoding $\tilde{C}_{est}$ with respect to the ideal aggregate chips for every simultaneous bit combination. Specifically this involves finding the bit combination that minimizes the Euclidean distance between the expected and estimated aggregate chips.

TABLE I
TRUTH TABLE RELATING SIMULTANEOUS DATA CHIPS $\tilde{D}$ TO THE RESULTING AGGREGATE CHIPS C AND THE FINAL CORRELATES X IN A THREE SATELLITE SCENARIO.

$\tilde{D}_0$	$\tilde{D}_1$	$\tilde{D}_2$	$\Rightarrow$	$\pm\tilde{C}(0)$	$\pm\tilde{C}(1)$	$\pm\tilde{C}(2)$	$\Rightarrow$	$\tilde{X}_0$	$\tilde{X}_1$	$\tilde{X}_2$
1	1	1		$\pm(1)$	$\pm(1)$	$\pm(1)$		1	1	1
−1	−1	−1		$\pm(-1)$	$\pm(-1)$	$\pm(-1)$		−1	−1	−1
−1	1	1		$\pm(3)$	$\pm(-1)$	$\pm(-1)$		−5	3	3
1	−1	−1		$\pm(-3)$	$\pm(1)$	$\pm(1)$		5	−3	−3
1	−1	1		$\pm(-1)$	$\pm(3)$	$\pm(-1)$		3	−5	3
−1	1	−1		$\pm(1)$	$\pm(-3)$	$\pm(1)$		−3	5	−3
1	1	−1		$\pm(-1)$	$\pm(-1)$	$\pm(3)$		3	3	−5
−1	−1	1		$\pm(1)$	$\pm(1)$	$\pm(-3)$		−3	−3	5

Consider three satellites with $\tilde{ML}_0 = (-1, 1, 1)$, $\tilde{ML}_1 = (1, -1, 1)$, and $\tilde{ML}_2 = (1, 1, -1)$. Suppose that the gains are equal and normalized $g_0 = g_1 = g_2 = 1$. By substituting the known $\tilde{ML}_i$ and every data combination $\tilde{D}$ into Equation 3, we construct the aggregate chips $\tilde{C}$ and thus the truth table given in Table I. Note that each data combination, along with its inverse, results in a unique aggregate chip sequence allowing the attacker to find the closest pair of rows to any received aggregate chips.

This method allows the attacker to determine only two possibilities for the simultaneously transmitted data bits with high probability, and with certainty in any system with an odd number of satellites and equal gains. The two possibilities stem from the uncertainty in the sign of $\tilde{PN}$, which can take only two possible values. Any additional uncertainties stem from from rare scenarios where other aggregate chip sequences are also plausible. A proof of this is given in Appendix C. We evaluate this decoder's performance in Section VI, where our results show high decoding performance in practice in systems with both odd and even numbers of satellites. It may be possible to construct a more efficient decoding algorithm which scales to larger numbers of satellites by not computing the entire truth table by instead directly solving the optimization problem in Algorithm 1. However, since the maximum likelihood algorithm is feasible in all scenarios specified by the final report (where up to 10 simultaneous transmissions are supported), we consider this out of scope [14, 15].

C. Pseudo-noise recovery

The Synchronization Data SD used to initialize the session is not protected against replay (cf. Section III-A). Therefore an attacker that recovers the pseudo-noise PN associated with a given SD can spoof without cracking the secret key, where only one satellite's data sequence is known, by replaying the SD header followed by their own data modulated with the recovered PN. The only way to truly recover the ideal PN is to gain a sufficiently low-noise recording such that each individual chip can be demodulated, which is likely infeasible.

Nevertheless, since the spreading codes are designed to be received in a noisy environment, the attacker is not required to recover the ideal PN; a noisy estimate suffices.

To recover this estimate, we consider the signal received by the attacker which is the transmitted signal from Equation 3 plus a channel noise component:

$$\tilde{sig}(t) = \tilde{PN}(t) \cdot \tilde{C}(t) + n(t) \quad \text{where} \quad n(t) \sim \mathcal{N}(0, N_0) \tag{7}$$

Assuming that the attacker successfully received the aggregate chips by the method in Section V-A and decoded the data as in Section V-B, then an estimate for all satellites' data sequences, $\tilde{D}_{est}$, can be calculated. The attacker can use this to construct $\tilde{C}_{est}$, an estimate of the aggregate chips $\tilde{C}$. Where $\tilde{C}_{est}$ is not zero, we can divide the eavesdropped signal $\tilde{sig}_e$ by it to recover the pseudo-noise estimate $\tilde{PN}_{est}$:

$$\tilde{PN}_{est}(t) = \begin{cases} \frac{\tilde{C}(t)}{\tilde{C}_{est}(t)} \cdot \tilde{PN}(t) + \frac{n(t)}{\tilde{C}_{est}(t)} & \text{if } \tilde{C}(t) \neq 0 \\ 0 & \text{if } \tilde{C}(t) = 0 \end{cases} \tag{8}$$

Note that PN is recovered, notwithstanding the noise term, when $\tilde{C}(t) = \tilde{C}_{est}(t)$. Now the attacker-chosen spoofed data sequence D_s can be modulated with the desired satellite ML sequences and $\tilde{PN}_{est}$ to form the spoofed signal $\tilde{sig}_s$. In terms of the resulting aggregate chips within the receiver:

$$\tilde{sig}_s(t) = \tilde{C}_s \cdot \tilde{PN}_{est} \tag{9}$$

$$= \begin{cases} \frac{\tilde{C}_s(t) \cdot \tilde{C}(t)}{\tilde{C}_{est}(t)} \cdot \tilde{PN}(t) + \mathcal{N}\left(0, \frac{\tilde{C}_s^2(t)}{\tilde{C}_{est}^2(t)} \cdot N_0\right) & \text{if } \tilde{C}(t) \neq 0 \\ 0 & \text{if } \tilde{C}(t) = 0 \end{cases} \tag{10}$$

The effect of this estimation on the final spoofed signal sig_s is twofold. First any errors from eavesdropping result in $\tilde{C}_{est} \neq \tilde{C}$, which introduces an error of $\frac{\tilde{C}}{\tilde{C}_{est}}$ in the $\tilde{PN}$ term. Second, the noise power is inversely proportional to the squared estimated aggregate chips $\tilde{C}_{est}(t)$, so that the louder the chips the better the spoofer's SNR. If $\tilde{C}_{est}^2(t) > \tilde{C}_s^2$ the effect is reduced noise and increased SNR, but if $\tilde{C}_{est}^2(t) < \tilde{C}_s^2$ the effect is increased noise. If instead $\tilde{C}_{est}(t) = 0$, which can only occur with an odd number of satellites communicating, then the eavesdropper is uncertain as to the phase of $\tilde{PN}(t)$ which is therefore unrecoverable in this position t.

We evaluate the performance of this method in Section VI-B and evaluate in which real-world contexts the noisy estimate suffices in Section VI-D.

D. Hybrid-optimized jamming

We finally show that, in addition to being broken against eavesdropping and spoofing attacks, the hybrid system enables lower-power denial of service through jamming as compared to a non-hybrid DSSS system. We consider a jammer which is interested in denying service to a subset of the concurrent satellites, which may be the entire set.

The key observation is that, whilst the correlation $\tilde{X}_i$ between $\tilde{sig}$ and $\tilde{ML}_i$ is used to identify the bits by observing the sign, certain $\tilde{X}_i$ are lower power than others. Despite this, each $\tilde{X}_i$ conveys the same amount of information. The attacker can therefore identify and selectively jam only these lower-power correlations to improve performance.

Specifically, the correlate values are denoted $\tilde{X}_i$ in the system model (see Section III), and their sign indicates the relevant data bit. From Table I we see that the relationship between the data $\tilde{D}$ and resulting values $\tilde{X}_i$ vary in magnitude (and therefore power) depending on which pair of aggregate sequences was transmitted. Assuming a Gaussian jammer model, the relationship between a given $\tilde{X}_i$ and its resulting bit error rate in the presence of noise N_0 can be calculated by considering the probability that the noise causes a deviation of at least $|\tilde{X}_i|$ from the original signal position:

$$BER = P[\mathcal{N}(0, N_0) > |\tilde{X}_i|] = \frac{1}{2}\text{erfc}\left(\frac{|\tilde{X}_i|}{\sqrt{2N_0}}\right) \tag{11}$$

where erfc is the Gaussian *complementary error function*.

Under a full knowledge jammer model, where sections of the data (e.g. fixed headers) are known beforehand, the attacker can identify low-power bits and focus the jamming signal on these. Without full knowledge, the jammer can instead be reactive and use the first part of each bit period of M chips to estimate the aggregate chip sequence using the method in Section V-A. Then in the latter part of the bit period, they can decide the jammer power level to transmit based on their objective. We note that the reactive method does not assume that the attacker knows any data bits beforehand, since identifying only a pair of possible data sequences is sufficient to know the power of each $\tilde{X}_i$.

We compare the resulting error rate in the context of the system to a typical Gaussian jammer in Section VI-C.

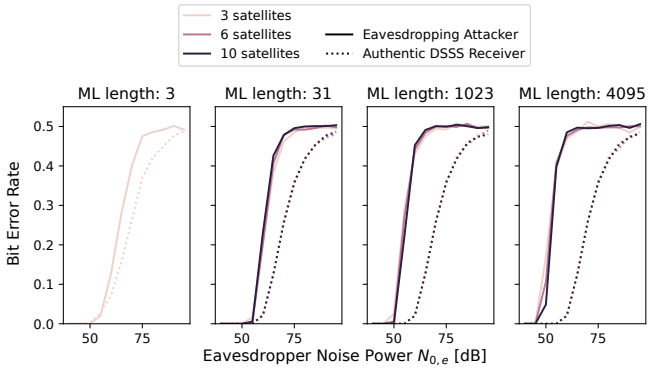
VI. Evaluation

We now proceed to evaluate the above methods for eavesdropping, spoofing, and jamming. Our primary objective is to compare the performance of the adversary, in terms of data successfully recovered/spoofed/jammed, to the performance of the authentic system. We then perform an end-to-end analysis where we account for the physical environment, including the path loss involved when the attacker and satellites are different distances from the ground station. All source code can be found at https://github.com/ssloxford/hybrid-crypto-spreading-code.

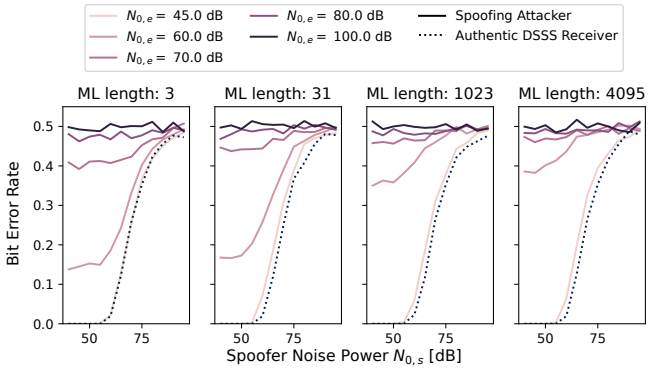
A. Eavesdropping

In this simulation we first generate the victim ML sequences by finding a maximum-length sequence of a given length, and performing unique bit-rotations. The victim data is given by modulating a random binary sequence for each satellite with its corresponding $\tilde{ML}$ sequence, and then taking the sum of each chip. This gives the aggregate chip sequence that the attacker is able to recover subject to the $\pm$ uncertainty introduced by pseudo-noise PN. Since the attacker takes the absolute value of each chip to recover the amplitude, we set all aggregate chips to have positive value.

We adopt a Gaussian noise model so that each aggregate chip is offset by a random amount dictated by a Gaussian



(a) Eavesdropping attacker, varying no. satellites.



(b) Spoofing attacker, varying $N_{0,e}$ for two satellites

Fig. 3. Comparison of the attacker compared to the authentic DSSS Receiver as the SNR varies. Low Bit Error Rate (BER) indicates attack success, 0.5 indicates decoding is no better than guessing. A number of different ML lengths are considered, with a long spreading sequence of $M = 2^{22}$. W.l.o.g. we set the waveform in both cases to have signal power 1.

random variable with mean 0 and variance N_0 which is the noise power. We assume w.l.o.g. that the signal has unit power 1, so that the Signal-to-Noise power Ratio (SNR) is $1/N_0$. Each aggregate chip is the result of averaging together M/N chips — the ratio between the length of the spreading sequence for a given bit, and the length of the MLs. Therefore the noise variance for each aggregate chip is given by taking the standard error of the mean, giving noise power per aggregate chip $N_0 \cdot \frac{N}{M}$.

For each aggregate chip combination, we perform Maximum-Likelihood decoding as described in Section V-B by finding the bit combination with the closest aggregate chip combination in Euclidean space. We compute the bit error rate (BER) by assuming that one of the bits within the combination is known, allowing all bits within the combination to be recovered with high probability, and averaging over many trials with randomized data. In the case that none of the bits are known, this is equivalent to the error rate of identifying the bits subject to a single sign uncertainty.

In Figure 3a we compare the bit error rate of the eavesdropper (solid line) to an authentic receiver (dotted line), under different ML lengths and as SNR increases. It can be seen that as the SNR increases, so both the eavesdropper and authentic receiver are able to recover the data sequence more effectively, with the authentic receiver achieving a higher performance. The decoder performance for systems with odd and even numbers of satellites is indistinguishable. Since the SNR is related to the path distance, and the distance to the satellite is so large, it is always possible for an attacker to recover the data by being sufficiently close to the ground station. We consider this analysis in detail in Section VI-D.

Interestingly the eavesdropper performs worse with respect to the authentic receiver as ML length increases. Whereas the authentic receiver directly correlates for each $\tilde{ML}$, which are designed for the minimum possible cross-correlation, the eavesdropper cannot do this without knowledge of $\tilde{PN}$, so instead compares the Euclidean distance. This distance is not necessarily maximized between bit combinations, leading to decreased performance for longer ML sequences.

In all of these plots we have picked a long spreading sequence of $M = 2^{22}$ as an example since it has been evaluated in a secure spreading code context, but the effect of a different value of M is only a constant x-axis offset [30]. This is because the equivalent noise power and therefore equivalent SNR scales linearly with respect to M/N [12].

B. Spoofing

In this simulation we consider an attack conducted in two stages: first eavesdropping to recover a pseudo-noise estimate associated with given synchronization data, and then spoofing using this estimate. We assume that the attacker transmits the spoofed message when no other communication is present [29]. We consider the hardest scenario where the attacker only targets a single satellite; this minimizes interference and therefore is the most robust to noise.

We first modulate a randomly-chosen victim data sequence with the satellite $\tilde{ML}$ sequences resulting in aggregate chips $\tilde{C}$. We then simulate eavesdropping with knowledge of a single bit sequence to recover the eavesdropper's estimate of the aggregate chips $\tilde{C}_{est}$, with respect to eavesdropper noise power $N_{0,e}$. The pseudo-noise estimate $\tilde{PN}_{est}$ is recovered, including errors from the eavesdropping stage, using Equation 8.

The attacker signal is then constructed by modulating the data sequence with both the $\tilde{ML}$ of a given satellite and $\tilde{PN}_{est}$. At the victim satellite, this signal is received with a further additive noise term $N_{0,s}$ representing the channel noise between the attacker and victim. Thus the final signal model is given by:

$$\tilde{sig}_s(t) = \tilde{C}_s \cdot \tilde{PN}_{est} + N_{0,s} \tag{12}$$

At the victim satellite, we decode the sequence by correlating $\tilde{sig}_s$ with the received signal $\tilde{PN} \times \tilde{ML}_i$. The average BER is calculated over many trials with randomized data.

In Figure 3b we compare the performance of the spoofer (solid line) to an authentic receiver (dotted line), and vary

the length of the ML sequence. The line color indicates the eavesdropper channel noise $N_{0,e}$. We consider a scenario of 2 satellites which allows us to compare across all ML lengths.

It can be seen that as $N_{0,e} \rightarrow -\infty$, the performance of the spoofing attacker approaches but never reaches the authentic receiver. This is to be expected as fundamentally the attacker must use a pseudo-noise estimate which is sometimes unrecoverable when $C_{est} = 0$ (see Equation 8) even in ideal noise conditions. As $N_{0,e}$ increases, so the error rate of the attacker degrades and approaches $BER \approx 0.5$. The length of the ML sequence is seen to have comparatively little impact on the overall performance.

We can understand the impact of errors introduced in the eavesdropping stage by comparing Figures 3a and 3b. Taking $N_{0,e} = 60$ as an example, as seen on the x-axis of Figure 3a, we see the Bit Error Rate increases as ML length increases. For instance, $BER \approx 0.45$ for ML length 4095. Comparing this to the $N_{0,e} = 60$ line in Figure 3b we see the resulting minimum Bit Error Rate $BER \approx 0.25$. The more robust direct correlation decoding performed at the receiver can make up for a significant proportion of errors introduced at the eavesdropping stage, assisting the attacker.

C. Jamming

Now we consider a jamming attack conducted against the *Secure Communication* phase of the hybrid system which takes advantage of the low-power correlates from Section V-D. We consider the *full knowledge* jammer model since it has been shown that exploiting the fixed, repeating header structure of many satellite protocols is sufficient to stealthily deny service [27]. We assume the worst-case condition where during known periods of communication the data sequence is identical for all satellites. The jammer transmits Gaussian-distributed symbols, and so is not required to be synchronized to the individual chips.

We simulate this scenario by applying Equation 11 which relates jammer noise power N_0 to Bit Error Rate (BER) for a single satellite i. The Gaussian jammer performance is found by taking the average bit error rate over all equally-likely aggregate chip sequences. The hybrid-optimized jammer performance is found by averaging over just the lowest power aggregate chip sequences.

In Figure 4 we plot the BER for these two jammer types, for a selection of different ML lengths. For low ML lengths the hybrid-optimized jammer significantly outperforms the Gaussian jammer by $\sim 10\,\mathrm{dB}$. This is because the minimum correlate power targeted by the hybrid-optimized jammer is proportionally significantly lower than the average correlate power targeted by the Gaussian jammer. As the ML length increases, so we see the performance of the jammer classes converge, as for long MLs there is relatively little difference between the minimum and average correlate powers.

D. End-to-end analysis

In the above analyses, we derive the performance of the attacker as compared to a baseline performance level in dB.

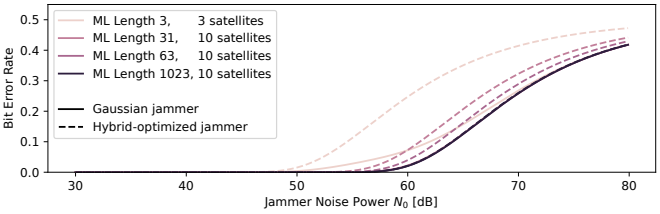


Fig. 4. Performance comparison of a Gaussian jammer (solid line) against the hybrid-optimized jammer (dashed line), for varying ML lengths. Higher Bit Error Rate (BER) indicates jammer success. Number of satellites chosen according to the maximium the ML supports, specifically $max(ML$ Length, $10)$.

To consider this for the real-world adversary scenarios in Figure 1, we now relate the performance in dB to the attacker distance d_a and satellite distance d_s derived from the well-known formula for free space path loss:

$$d_a = d_s \cdot 10^{-\frac{\Delta G\ [\mathrm{dB}]}{20}} \tag{13}$$

d_a/d_s is the distance relative to the satellite that the attacker has to account for in order to make up the performance loss of $\Delta G[\mathrm{dB}]$. To evaluate the attacker's capability in terms of performance loss, we assume that like typical onboard antennas, the attacker antenna is also omnidirectional. Clearly a more sophisticated attacker could make up for this loss using a higher gain antenna, or by being present in antenna sidelobes, but for generality we do not account for differences in antenna gain patterns.

1) Eavesdropping: In Section VI-A we found that an eavesdropping receiver without knowledge of the secret key suffers degraded performance as compared to an authentic receiver which knows the key and can directly correlate for the spreading sequence. From Figure 3a we see that the performance degrades by no more than $< 20\,\mathrm{dB}$ in all tested cases. Assuming that the thermal noise level for both the attacker and victim receiver are identical, we can apply free space path loss using Equation 13. We find that in this worst case, the attacker must be $1/10^{\text{th}}$ the distance to the ground station in order to make up the loss. This makes eavesdropping highly possible in the ground-based and in-beam scenarios of Figures 1a and 1b, but extremely hard in the satellite scenario of Figure 1c except if substantially different orbits are considered (e.g., a LEO satellite intercepting GEO transmissions).

2) Spoofing: In Section VI-B we determined that there are two key factors in a successful spoofing attack: that both the eavesdropping $SNR_{0,e}$ between the ground station, and the spoofing $SNR_{0,s}$ between the attacker and satellite, be sufficiently high. Since the required spoofing SNR can be achieved with an additional amplifier to make up for the path loss, we focus on the required spoofing SNR.

We can evaluate this by considering the Gaussian jammer performance in Figure 4, which is equivalent to the eavesdropper performance in an AWGN channel. In this plot, the number of satellites is maximized for each ML, giving the worst-case performance with maximum inter-satellite interference.

Depending on the system, we see that to achieve $BER \to 0$, the acceptable range for $N_{0,e}$ varies within $50 \ldots 60\text{dB}$, which corresponds to an SNR of $-60 \ldots -50\text{dB}$.

We now compare this to Figure 3b, where we see that in order to achieve a spoofed $BER \approx 0$, the equivalent SNR needs to be $-45\,\text{dB}$ in the worst case. The difference between the baseline and spoofer performance reveals a performance gap of $15\,\text{dB}$, which by applying Equation 13 can be made up for if the attacker is a factor of 0.178 of the distance to the ground station, which is approximately $\frac{2}{10}^{\text{th}}$ the distance.

The result is that the attacker can receive a pseudo-noise estimate which is sufficient to spoof at twice the distance from the ground station as is required to successfully eavesdrop. This is surprising, but is due to two aspects. First, the victim satellite uses knowledge of the secret key to directly correlate for the data which is a higher performance receiver method than the eavesdropper can employ. Second, the spoofer uses the strongest possible signal and uses the pseudo-noise estimate to encode for one satellite only, making it easier for the satellite to receive.

3) Jamming: In Section VI-C we saw that the performance of the jammer can be substantially improved by targeting the low-power correlates, and that this has the greatest effect as compared to a Gaussian jammer for low ML lengths. The performance increase requires synchronization only to the bits, not to the chips. This performance increase is up to $10\,\text{dB}$ and can be realized either as a reduction in the jammer's transmit power or a decrease in distance as compared to the ground station-satellite link. Applying Equation 13 we see that the jammer can, at the same power level as the ground station, jam the satellite from $3.16\times$ the distance.

VII. Countermeasures

So far our study has shown that the hybrid scheme studied by ESA provides few of the security features intended by its design. Furthermore, the protocol can be broken even with low capabilities; knowledge of only satellite's data sequence reveals all the others under eavesdropping, no synchronization is required under spoofing, and only bit-level (and not chip-level) synchronization is required under jamming. Using our results, we make a number of insights about a more secure construction of the spread spectrum system.

A. Non-hybrid spread spectrum/per-satellite keying

As we saw in Section V, the reuse of the same cryptographic sequence PN for all satellites results in mutual information about the data sequences in the aggregate chips being present in the output. This is the basis for eavesdropping and hybrid-optimized jamming attacks. One mitigation approach is to abandon the hybrid system entirely and use a different pseudorandom spreading sequence per satellite, at the cost of up to $\sim 30\,\text{dB}$ of performance [12]. Future work should consider alternative protocols which provide both good security and multiple-access properties.

B. Preventing synchronization data reuse

Fundamentally spoofing can occur whenever the PN sequence for a particular session can be generated by the attacker. Whilst PN cannot be generated without knowledge of the secret key, we have shown in Subsections V-C and VI-B that a good enough PN estimate can be recovered with prior knowledge of the data. Whilst this is particularly effective when chained with our eavesdropping techniques, it is possible in the more general case if the protocol is sufficiently predictable. It is therefore essential that PN reuse is prevented.

To account for this, the *Synchronization Data* (SD) transmitted at the start of the session (as described in Section III) must be protected against reuse. This will involve a cryptographic signature covering both the SD and freshness information in the form of e.g. a timestamp. By setting a maximum acceptable tolerance between the SD timestamp and the satellite's clock, which is typically no more than a few seconds out from the ground station clock, spoofing outside the tolerance can be prevented. If sufficiently tight, the tolerance should make the timing for jamming, estimating PN, and retransmitting infeasible. Future work should consider the important security/robustness tradeoff in the appropriate selection of the tolerance and its relationship to distance-bounding protocols.

C. Cryptographic scrambling

Since recovery of the PN sequence depends on prior knowledge of the data, a natural countermeasure is to ensure that the data sequence cannot be estimated beforehand. Whilst an independent COMSEC security mechanism could provide this, the PN could still be deduced during known idle sequences which are unprotected by COMSEC methods.

To enhance this, a shared key could be used to derive a secure, per-session and per-satellite cryptographic scrambling sequence, which is often known as "Bulk Security'. This can be undone at the satellite, and has the effect of making the data unpredictable to an attacker, in particular eliminating the effect of elements such as predictable headers and all-zero idle sequences. This method mitigates spoofing the entire message since the PN sequence cannot easily be derived, but individual bits can still be flipped by the attacker. Eavesdroppers now cannot derive the data sequences, but side-channel information such as the number of simultaneously transmitting satellites is still available. Hybrid-optimized jamming, which relies only on mutual information, is not mitigated. As a result, cryptographic scrambling is only a partial mitigation.

VIII. Conclusion

In this work we have demonstrated that the hybrid CDMA/Cryptographic Direct Sequence Spread Spectrum (DSSS) scheme studied by the European Space Agency is fundamentally broken against low-capability adversaries. Whereas the system is intended to provide the multiple-access properties of CDMA with the security properties of cryptographic DSSS, we have shown and evaluated realistic attacks affecting data secrecy, authenticity, and availability.

In particular, due to reuse of the cryptographic pseudo-noise sequence, an eavesdropper can recover the entire data sequence of all satellites in the system when any one satellite's data sequence is known with high probability (made easier through predictable all-zero idle periods). Spoofing is possible since a sufficient estimate of the pseudo-noise sequence can then be recovered and reused owing to a lack of freshness guarantees in session keying. These attacks are possible in any scenario where the attacker can afford to be $10\times$ closer to the ground station than the satellite in the worst case, opening the system to attackers in either the ground station sidelobes or main beam. Finally we introduce a hybrid-optimized jammer which exploits the pseudo-noise reuse to deny service at up to $10\,\mathrm{dB}$ lower power than a typical Gaussian jammer.

These results have serious implications for ongoing developments in the ETSI standardization of cryptographic spreading codes for satellite applications being driven by ESA.

References

[1] 2017. ETSI EN 301 926 V1.3.1: Satellite Earth Stations and Systems (SES); Radio Frequency and Modulation Standard for Telemetry, Command and Ranging (TCR) of Communications Satellites. Standard. (Oct. 2017).

[2] 2011. CCSDS 415.1-B-1: Data Transmission and PN Ranging for 2 GHz CDMA Link via Data Relay Satellite Recommended Standard. Recommended Standard. (Sept. 2011).

[3] 2025. Space Frequency Coordination Group (SFCG). Website. (2025). https://www.sfcgonline.org/home.aspx.

[4] J. L. Massey. 1969. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, IT-15, 1, (Jan. 1969), 122–127. DOI: 10.1109/TIT.1969..

[5] Gilles Burel and Celine Bouder. 2000. Blind Estimation of the Pseudo-Random Sequence of a Direct Sequence Spread Spectrum Signal. In *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No. 00CH37155)*. Vol. 2. IEEE, 967–970.

[6] Saeed Mehboodi, Mahmoud Farhang, and Ali Jamshidi. 2016. Maximum likelihood estimation of pseudo-noise sequences in non-cooperative direct-sequence spread-spectrum communication systems. In *2016 24th Iranian Conference on Electrical Engineering (ICEE)*. IEEE, 119–123.

[7] Bin Shen and Jian-xin Wang. 2017. Chip rate and pseudo-noise sequence estimation for direct sequence spread spectrum signals. *IET Signal Processing*, 11, 6, 727–733.

[8] J. L. Massey and T. Mittelholzer. 1993. Welch's Bound and Sequence Sets for Code-Division Multiple-Access Systems. In *Sequences II: Methods in Communication, Security and Computer Sciences*. R. Capocelli, A. De Santis, and U. Vaccaro, (Eds.) Springer, Heidelberg and New York, 63–78.

[9] Zahoor Ahmed, J. P. Cances, and V. Meghdadi. 2008. Cryptographic Spread Spectrum Relay Communication. In *2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*, 588–591. DOI: 10.1109/NGMAST.2008.45.

[10] Thales-Alenia Space. 2012. Deep Space and Secure Transponders. Sales Sheet. (June 2012). Retrieved July 16, 2025 from https://web.archive.org/web/20250121120135/https://www.thalesgroup.com/sites/default/files/database/d7/asset/document/Deep_Space_Secure_Transponders.pdf.

[11] 2018. ETSI TR 103 956 V1.1.1: Satellite Earth Stations and Systems (SES); Technical Analysis for the Radio Frequency, Modulation and Coding for Telemetry Command and Ranging (TCR) of Communications Satellites. Technical Report. (Dec. 2018).

[12] G. Fittipaldi. 2021. Cryptographic Pseudo-Noise Codes and Related Acquisition Techniques for Direct-Sequence Spread Spectrum Transponders. Tech. rep. RPT-RFP-ESA-00013-AASI. Technical Report. European Space Agency, (July 2021).

[13] J. Massey. 2010. Code Selection & Trade-Offs. Tech. rep. Contract No. 22369/09/NL/JK, TASI Subcontract 010909/022/EQSI/RG. Issue 2. European Space Research and Technology Centre (ESTEC), (Mar. 2010).

[14] L. Simone, G. Fittipaldi, J. Massey, S. Vono, and V. Schena. 2011. Cryptographic Pseudo-Noise Codes and Related Acquisition Techniques for Direct-Sequence Spread Spectrum Transponders: Study Report. Tech. rep. RPT-RFP-ESA-00013-AASI. Technical Report. European Space Agency, (July 2011).

[15] L. Simone and G. Fittipaldi. 2012. Cryptographic Pseudo-Noise Codes and Related Acquisition Techniques for Direct Sequence Spread Spectrum Transponders: Final Presentation. In Technical Presentation. Radio-Communication Equipment Dept. – CCEL. Rome, Italy.

[16] Roberto Garello, Monica Visintin, Riccardo Schiavone, Alessandro Compagnoni, and Carla Fabiana Chiasserini. 2025. AES and Mixed AES/Gold Spreading Sequences for Satellite Uplink Code Division Multiplexing. *IEEE Transactions on Communications*, 1–1. DOI: 10.1109/TCOMM.2025.3554679.

[17] Anne Martin, Yeashfi Hasan, and R Michael Buehrer. 2013. Physical Layer Security of Hybrid Spread Spectrum Systems. In *2013 IEEE Radio and Wireless Symposium*. IEEE, 370–372.

[18] Frank Hermanns. 2004. Cryptographic CDMA code hopping (CH-CDMA) for signal security and anti-jamming. *EMPS 2004*.

[19] Lawrence Young, Thomas Meehan, and Jess B. Thomas. 2003. P-Code-Enhanced Encryption-Mode Processing of GPS Signals. Tech. rep. NASA Tech Briefs, p. 46. NASA's Jet Propulsion Laboratory, Pasadena, California, (Mar. 2003).

[20] Malte Lenhart, Marco Spanghero, and Panagiotis Papadimitratos. 2021. Relay/replay attacks on GNSS signals. In *Proceedings of the 14th ACM conference on security and privacy in wireless and mobile networks*, 380–382.

[21] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the requirements for successful GPS spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security*, 75–86.

[22] Michael Spuhler, Domenico Giustiniano, Vincent Lenders, Matthias Wilhelm, and Jens B. Schmitt. 2014. Detection of Reactive Jamming in DSSS-based Wireless Communications. *IEEE Transactions on Wireless Communications*, 13, 3, 1593–1603. DOI: 10.1109/TWC.2013.013014.131037.

[23] Wang Hang, Wang Zanji, and Guo Jingbo. 2006. Performance of DSSS against repeater jamming. In *2006 13th IEEE International Conference on Electronics, Circuits and Systems*. IEEE, 858–861.

[24] Daniel Moser, Vincent Lenders, and Srdjan Capkun. 2019. Digital Radio Signal Cancellation Attacks: An experimental evaluation. In *Proceedings of the 12th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 23–33. http://lenders.ch/publications/conferences/wisec19_1.pdf.

[25] Edd Salkield, Sebastian Köhler, Simon Birnbach, and Ivan Martinovic. 2024. Security Risks of Adaptive Coding and Modulation in Space Systems. In *2024 Security for Space Systems (3S)*. IEEE, 1–10.

[26] Marco Baldi, Franco Chiaraluce, Roberto Garello, Nicola Maturo, I Aguilar Sanchez, and Stefano Cioni. 2015. Analysis and performance evaluation of new coding options for space telecommand links - part II: jamming channels. *International Journal of Satellite Communications and Networking*, 33, 6, 527–542.

[27] Edd Salkield, Sebastian Köhler, Simon Birnbach, Martin Strohmeier, and Ivan Martinovic. 2025. SpaceJam: Protocol-aware Jamming Attacks against Space Communications. In *18th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 160–171.

[28] 2021. TC SYNCHRONIZATION AND CHANNEL CODING. CCSDS. (July 2021). Retrieved May 14, 2025 from https://ccsds.org/wp-content/uploads/gravity_forms/5-448e85c647331d9cbaf66c096458bdd5/2025/01//231x0b4e1.pdf.

[29] Edd Salkield, Marcell Szakály, Joshua Smailes, Sebastian Köhler, Simon Birnbach, Martin Strohmeier, and Ivan Martinovic. 2023. Satellite Spoofing from A to Z: On the Requirements of Satellite Downlink Overshadowing Attacks. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 341–352.

[30] Lorenzo Simone, G. Fittipaldi, and Ignacio Aguilar Sanchez. 2011. Fast acquisition techniques for very long PN codes for On-Board Secure TTC transponders, (Nov. 2011), 1748–1753. DOI: 10.1109/MILCOM.2011.6127563.

APPENDIX

A. Acknowledgments

Sebastian was supported by the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK Intelligence Community Postdoctoral Research Fellowships programme. Simon was supported by the Government Office for Science and the Royal Academy of Engineering under the UK Intelligence Community Postdoctoral Research Fellowships scheme. We would like to acknowledge the useful feedback from and discussions with ESA throughout the course of this work.

B. Responsible disclosure

These vulnerabilities affect ESA project *AO/1-5940/08/NL/JK* which is publicly available. We disclosed our findings via ESA to the authors of the study, Thales-Alenia Space Italy, who confirmed that the technique has not been implemented in their products.

C. Pairwise uniqueness of the decoder

The decoder described in Section V-B relies on mapping the received aggregate chips $\tilde{C}$ to the bit sequences D which could have caused them. Since the effect of the PN sequence is to randomize the sign, D and $-D$ are indistinguishable. Here we show that, other than this single sign, the mapping is unambiguous for odd numbers of satellites when the gains $g_s = 1$, but are ambiguous in rare cases otherwise. These ambiguous cases theoretically affect the performance of the decoder, but the results from Section VI-A show that they make little difference in practice.

1) Odd number of satellites with all $g_s = 1$: Consider two data sequences $\tilde{D}$ and $-\tilde{D}$ of all simultaneous satellites, which result in aggregate chip sequences $\tilde{C}$ and $-\tilde{C}$ respectively. Suppose that there exists a third distinct data sequence $\tilde{D}'$ which results in aggregate chip sequence $\tilde{C}'$ such that $|\tilde{C}| = |\tilde{C}'|$. By the construction of C from Equation 3, and by considering only a single bit period, this can be written with respect to a set S, the satellites whose bits differ between D and D':

$$\sum_{s\in S} d_s \times \tilde{\mathbf{ML}}_s + \sum_{s\in S^{\complement}} d_s \times \tilde{\mathbf{ML}}_s = \mathbf{v}(-\sum_{s\in S} d_s \times \tilde{\mathbf{ML}}_s + \sum_{s\in S^{\complement}} d_s \times \tilde{\mathbf{ML}}_s) \quad (14)$$

The RHS is $\tilde{C}'$ which has all bits in S flipped. $\mathbf{v}$ is a vector of $+1$ and -1, and accounts for fact that the aggregate chips are considered equivalent regardless of the sign of each row. We now consider the cases where S contains an even or an odd number of satellites with flipped bits.

Even Case: Consider just a single row in the vectors of Equation 14. If v takes value -1 in any row then the equation simplifies in that row to $\sum_{s\in S^{\complement}} d_s \times \tilde{ML}_s = 0$. However this is not possible since there are an odd number of satellites in the complement of S, $S^{\complement}$. Therefore v must take value $+1$ in every row, and so the equation simplifies to $\sum_{s\in S} d_s \times \tilde{\mathbf{ML}}_s = 0$. This is impossible since all the maximum length vectors ML_s are linearly independent[5].

Odd Case: This case is equivalent: v cannot take value $+1$ in any row since there are an odd number of satellites in S, so the equation also simplifies to $\sum_{s\in S} d_s \times \tilde{\mathbf{ML}}_s = 0$ for all rows which is similarly impossible.

2) Other cases: When there are an even number of satellites, or the gains g_s are not all equal, it is now possible to construct S to satisfy Equation 14. Consider the ML sequence generated from a length 4 shift register $\mathbf{m} = [1,1,1,1,-1,1,-1,1,1,-1,-1,1,-1,-1,-1]$. Let S consist of the set of all satellites with $\tilde{\mathbf{ML}}$ equal to $\mathbf{m}$ left bit shifted by indices $0,1,2,7$ and $S^{\complement}$ by $4,5,7,9$. The resulting aggregate sequence does not change regardless of whether $d_s = 1$ or $d_s = 0$ for $s \in S$. This is possible since every row that is non-zero in $\sum_{s\in S} d_s \times \tilde{\mathbf{ML}}_s$ is zero in $\sum_{s\in S^{\complement}} d_s \times \tilde{\mathbf{ML}}_s$.

Algorithm 1 Eavesdropping Decoder Optimization

EAVESDROP$(\mathbf{b}, \mathbf{ML}, \mathbf{g}) \rightarrow (\mathbf{D}^*, \mathbf{PN}^*)$

Constants

$b_1, \ldots, b_N$	Received aggregate chips
$ML_1, \ldots, ML_n$	Satellite ML sequences
$g_1, \ldots, g_n$	Satellite gains

Variables

$D_1, D_2, \ldots, D_n$	Data chip values
$PN_1, \ldots, PN_N$	Cryptographic pseudo-noise
$e_1^+, e_1^-, \ldots, e_N^+, e_N^-$	Error terms to minimize

Key principle: Find data D_i and pseudo-noise PN_i that minimize distance between received and expected chips.

Objective:

Minimize $Z = e_1^+ + e_1^- + \ldots + e_N^+ + e_N^-$

Key Constraints:

$g_1 ML_1[1] D_1 PN_1 + \ldots + g_n ML_n[1] D_n PN_1 + e_1^+ - e_1^- = b_1$

$\ldots$

$g_1 ML_1[N] D_1 PN_N + \ldots + g_n ML_n[N] D_n PN_N + e_N^+ - e_N^- = b_N$

Bounding Constraints:

$-1 \le D_1, \ldots, D_n, PN_1, \ldots, PN_N \le 1$

$e_1^+, e_1^-, e_2^+, e_2^- \ge 0$

Formulation of the Equation 3 system as an optimization problem, which if solved approximately and/or efficiently could allow accurate decoding with large numbers of satellites. The problem bears resemblance to a linear program, except for the multiplication of variables D_i and PN_j in each of the *Key Constraints*, making it in class *MINLP* (Mixed Integer Non-Linear Program). Solving for D given a candidate PN is efficient since the system reduces to a linear program. Initial analysis indicated that this more general problem is likely non-convex.

[5]This was confirmed numerically for all sequences with up to length 11 registers. A proof in the general case is considered out of scope for this paper.