A FAST INFORMATION THEORETICALLY SECURE RADIO COMMUNICATION PROTOCOL BASED ON GNSS POSITIONING

Sumio Morioka[1], Satoshi Obana[2] & Maki Yoshida[3]

[1]Interstellar Technologies Inc., Tokyo-East21 6F, 6-3-2 Toyo, Koto-ku, Tokyo 135-0016 Japan, (sumio.morioka@istellartech.com, tel. +81-3-6666-5711)
[2]Hosei University, 3-7-2 Kajino-cho, Koganei-shi, Tokyo 184-8585 Japan, (obana@hosei.ac.jp, tel. +81-42-387-6036)
[3]NICT, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795 Japan, (maki-yos@nict.go.jp, tel. +81-42-327-7429)

Abstract

Space launch vehicles and satellites have desired a highly secure and reliable radio communication protocol to protect their critical uplink and downlink. We have proposed a protocol that guarantees information theoretic security, which cannot be broken even if an attacker has unlimited computing power, unlike computational security used in most existing cryptographic systems. The proposed protocol uses GNSS time information to establish a reliable key synchronization between spacecraft and ground station and to eliminate the risk of a replay attack. While we have successfully conducted some flight tests on our sounding rocket MOMO and given formal proof that the proposed protocol is secure and robust, the maximum throughput is limited to approximately 100kbps because a sufficiently large channel-occupation ratio should be kept constantly. In this paper, we have modified the proposed protocol so that the throughput is improved to an order of 10Mbps by incorporating the use of GNSS position data into the key synchronization mechanism.

Keywords: NewSpace, Wireless Communication, Information Theoretic Security, Key Synchronization, GNSS

1. Introduction

In the recent NewSpace era, many rockets and satellites have been launched on various academic or commercial missions, such as constructing satellite constellation networks. Radio communication is a fundamental function of these spacecraft, and it is critical to establish high-level security as much as possible. For example, no impersonation attack nor tampering attack against critical uplink commands of spacecraft should not be successful in maintaining public safety, and interception of downlink transmissions that involve status information of spacecraft and highly variable mission data is undesirable.

Our primary motivation is to establish a high security level in the radio communication systems of space rockets and satellites. In [1, 2], we have identified and enumerated basic requirements in spacecraft uplink and downlink and proposed a new protocol that guarantees the highest theoretical security level known as information theoretic security [3]. An outstanding feature of information theoretic security is that it cannot be broken even if an attacker has unlimited computing power. This is unlike computational security, which is used in most existing cryptographic systems and is based on current and future computer capability limitations. To the best of the authors' knowledge, no spacecraft radio communication system achieves information theoretic security, even though a standardization work of space data link (mainly for satellites) has been underway in CCSDS [4].

In the proposed protocol, a famous one-time pad technique is used where many secret keys are shared between the spacecraft and the ground station, and the used key is changed and discarded in every packet transmission. Establishing key synchronization between sender and receiver is critical to ensure that the same key is applied to the corresponding packet, and our basic idea is to use

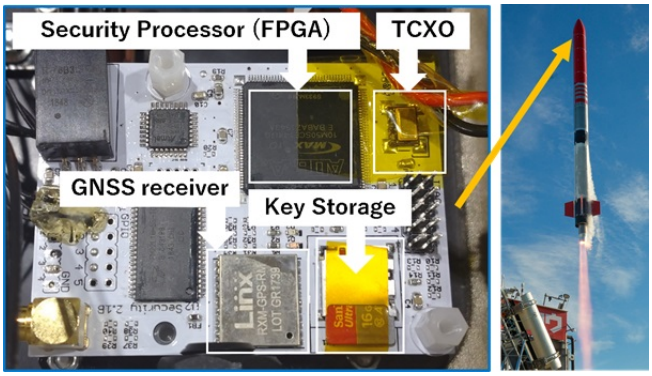

Figure 1 – Flight test of the proposed protocol on our sounding rocket MOMO [5–7]

precise GNSS-time information for computing index (address) value so that the same position of key storage devices is read in spacecraft and ground station.
While we have conducted several flight demonstration tests [5] of a prototype version of the protocol on our sounding rocket MOMO [6, 7] (Figure 1) and no significant communication error is observed, we decided to enhance the reliability of the protocol before the use in our satellite launcher ZERO [8], which is currently under development. Because the protocol used a state variable (i.e., data stored in the processor's main memory) to implement a countermeasure to replay attack, there was a risk of permanent communication loss following the destruction of the state variable caused by some H/W anomalies, such as soft errors and/or momentary power shutdown.
Therefore, in [1, 2], we have modified our protocol so that detection of a replay attack can be done by using GNSS time information without using any state variable. We have also provided formal proof of the detection algorithm. However, the protocol's maximum throughput has been limited to approximately 100kbps instead because a sufficiently large channel-occupation ratio should be maintained constantly to eliminate any empty channel slots that attackers' replay packets can use.
In this paper, we have modified the proposed protocol so that the throughput is improved to an order of 10Mbps by incorporating the use of GNSS position data into the key synchronization mechanism. We have confirmed that the throughput value is achievable on conventional consumer-grade embedded processors such as FPGAs.
This paper is organized as follows. In Sections 2 through 4, we overview the system requirements, security model, and the previous versions of our proposed protocols P1−P6 in [1, 2], respectively. In Section 5, we explain the modified fast protocol P7, estimation of the throughput, and prototype implementation results.

2. System Requirements

We first recall a general architecture and afterward security requirements [1, 2, 5].

2.1 System Participants

There are four main participants in the target communication system.

Ground Station (GS): Ground station(s) communicate with spacecraft by radio communication channels. Typically, a station uses multiple channels whose radio frequencies are different. Each channel is assigned to a single purpose and unidirectional (uplink or downlink) data transmission.[1] Every station has facilities such as radio transmitters, receivers, antennas, and power supply and can carry out operations with minimum support from outside. We assume their wireless communication systems are isolated from outside networks.

Small Launch Vehicle (SLV): The launch vehicle is a spacecraft to put satellites into Earth's orbit. We do not limit the target orbit or the GN&C method. The typical lifetime, or flight duration, of a launch vehicle is less than a few hours. A launch vehicle has multiple radio channels for communicating with ground stations.

[1] The proposed protocol aims to be used in one channel.

Small Satellite (SS): A small satellite is a spacecraft that conducts various commercial or academic missions in an Earth orbit. A satellite also has multiple radio channels. In this paper, we assume that the primary purpose of the communication channels is to link satellites with ground stations. Yet the use of satellite-to-satellite communication will be expanded in the future.

GNSS Satellites (GNS): Some recent spacecraft have onboard GNSS receivers to get time and location information. For example, some launch vehicles use a GPS/INS algorithm (a sensor fusion of GPS and IMU) in their GN&C system.

2.2 Security Requirements against Intentional Attacks

We first examine security requirements for the communication system. An entity that behaves maliciously in communication among the above entities (called an attacker) has the following capabilities but cannot physically access GS, SLV, SS, and GNS.

Passive attack (eavesdropping/inference): An attacker can detect the frequency of radio waves emitted by GS, SLV, SS, and GNS and can intercept and record signals. Furthermore, the modulation method and data contents can be estimated from the signals when plaintexts are transmitted. This allows the attacker to analyze control information and eavesdrop on high-value mission data.

Active attack (impersonating/tampering): An attacker can use estimated information and send a recorded signal, i.e. replay attack, or a tampered signal with sufficient strength. If control information/mission data could be tampered and a sender can be impersonated, then it would be impossible to carry out missions including flight termination/continuation control.

Destructive attack (jamming): An attacker can send noise signals with sufficient strength. This causes a communication failure and makes it impossible to carry out missions due to losing flight termination/continuation control.

In this paper, we focus on passive and active attacks. Countermeasures against destructive attacks, including interference with signal reception from positioning satellites, remain future issues. Security, in a narrow sense against passive and active attacks, is twofold: confidentiality and integrity, but in recent years, availability has also been contained in security. Thus, it is desirable to satisfy the following three security properties.

Confidentiality: In any uplink transmission, any information generated by GS can be obtained only by SLV/SS intended by GS. The same applies to downlink transmissions.

Integrity: In any uplink transmission, any information generated by GS can be confirmed on SLV/SS side that the source is GS and has not been tampered by others. The same applies to downlink transmissions.

Availability: In any uplink transmission, SLV (or SS) can obtain information sent by GS during a predetermined period within a predetermined time. The same applies to downlink transmissions. Availability is strongly required for uplink transmissions from GS to SLV. In an emergency, GS must (repeatedly) send a flight termination command to ensure it is executed within a predetermined short time.

Note that the security of communication between GSs, physical security of GSs, and physical security before the launch of SLV/SS are out of the scope of this paper, assuming that different mechanisms guarantee them. In addition, regarding information transmitted from GNS and used by other entities, we assume that security is guaranteed by mechanisms such as Galileo Open Service Navigation Message Authentication.

2.3 Robustness Requirements in Flight Environment

We examine robustness requirements based on the communication with SLV/SS in flight and the use of hardware equipment in outer space. Because the communication system is the only means of accessing a spacecraft in flight, if the communication system becomes unavailable, then the spacecraft will be lost. Maintaining a spacecraft has higher value and priority than satisfying security requirements. Therefore, it is essential to recover from communication anomalies and continue operating the spacecraft. We also examine robustness requirements based on the possibility that the spacecraft will be captured by a third party after the operation of the spacecraft is completed.
Thus, it is desirable to satisfy the following three robustness properties.

Robustness against unpredictable communication delay: There is little difficulty in designing a communication protocol if the communication delay between SLV/SS and GS is a known constant value. However, the actual communication delay is difficult to predict precisely. Therefore, it is necessary to maintain synchronization between sender and receiver and prepare a recovery mechanism from synchronization failures.

Robustness against temporal malfunction: Detecting errors and/or restoring data is required to normalize operations in the event of temporary or permanent data loss, damage, or malfunction of hardware caused by cosmic rays. The effects of soft errors and hard errors can accumulate and become apparent during the medium—to long-term operations of SS, as opposed to the short-term operations of SLV.

Robustness against recovering spacecraft: It is possible that SLVs that have finished their lifetime can be recovered, and the stored information (key storage, etc.) can be directly read by a third party in the future. However, it is still difficult with the current technology. The confidentiality of communications before the time of recovery vehicles must be maintained (so-called forward security). Integrity and availability are not particularly required because the data processing has been completed at the time of vehicle recovery. On the other hand, as mentioned above, physical security is guaranteed in the vehicle assembly facility and GS, and it is impossible to access the spacecraft during flight physically. Thus, it is assumed that no information will be leaked from the spacecraft during flight.

3. Security Model

In this section, we briefly recall the security model for secure communication between GS and SLV/SS [1, 2, 5]. More concretely, we show our formal definition of availability, confidentiality, and integrity for a secure communication system.

3.1 System Overview

We first explain how to model queries to GNS and the wireless communication between two entities (GS and SLV/SS) in a target system.

Model of GNSS enquiries: We model enquiries to GNS as oracle access to $\mathscr{O}_{\mathsf{GNS}}$ where $\mathscr{O}_{\mathsf{GNS}}$ is the oracle that answers the absolute time. We also introduce a parameter jit_{GNS}, which represents the maximum difference between the answer and the absolute time.

Model of Wireless Communication: We model the wireless communication between GS and SLV/SS as read/write access to the global tape, where the global tape consists of semi-infinite cells. We use the notation $\mathsf{C}[i:j]$ to denote the contents of the i-th cell through the j-th cell of the global tape C. In the proposed model, there exist four kinds of global tapes $\mathsf{C}_U^{\mathsf{GS,SLV}}$ (uplink communication channel from GS to SLV), $\mathsf{C}_D^{\mathsf{SLV,GS}}$ (downlink communication channel from SLV to GS), $\mathsf{C}_U^{\mathsf{GS,SS}}$ (uplink communication channel between GS and SS), and $\mathsf{C}_D^{\mathsf{SS,GS}}$ (downlink communication channel between SS and GS). Hereafter, we use the notation $\mathsf{Alg}_{\mathsf{C}}^{\mathscr{O}}$ to denote an algorithm Alg which queries to an oracle $\mathscr{O}$ and access (read and write) to a global tape C.

Next, we explain the algorithms defined in our proposed model. The proposed model consists of three algorithms Gen, $\mathsf{Snd}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}$, and $\mathsf{Rcv}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}$ which denote key generation algorithm, data sending algorithm, and data receiving algorithm, respectively.
Data sending algorithm $\mathsf{Snd}_{\mathsf{C}}$ and data receiving algorithm $\mathsf{Rcv}_{\mathsf{C}}$ are equipped with their unidirectional head, which moves with the throughput corresponding to the bit rate of the underlying communication channel. $\mathsf{Snd}_{\mathsf{C}}$ generates a protected (i.e., encrypted and authenticated) transmission data $\{0,1\}^*$ and writes to the tape C. When no data is sent through the channel, the random sequence $\{0,1,\perp\}^*$ is automatically written to C where $\perp$ corresponds to the signal, which is difficult to determine whether it is 0 or 1. $\mathsf{Rcv}_{\mathsf{C}}$ reads transmission data from the communication channel and processes (i.e., decrypt and verify the authenticity) the transmission. In defining each algorithm, we use notations $\mathscr{K},\mathscr{S},\mathscr{M}$ to denote the set of keys, plaintexts, and transmission data.

Definition of Gen**:** the algorithm takes L representing the total number of transmission data communicated during the entire life cycle, ℓ representing the bit length of plaintext, and security parameters, $\varepsilon_{\mathsf{cor}},\varepsilon_{\mathsf{sec}},jit$ with respect to correctness, and security, and jitter with inputs, and outputs shared key $k \in \mathscr{K}$ between the sender and the receiver.

Definition of $\mathsf{Snd}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}$**:** the algorithm takes a shared key $k \in \mathscr{K}$, a fixed length plaintext $s \in \mathscr{S}$ as inputs, and outputs a transmission data $m \in \mathscr{M}$ to be written to the tape C through the head of Snd algorithm.

Definition of $\mathsf{Rcv}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}$**:** the algorithm takes a shared key $k \in \mathscr{K}$ with input and obtains transmission data from a global tape C. It then outputs $s' \in \mathscr{S}$ if the transmission data is verified as authentic or outputs $\perp$ otherwise. We should note that Snd and Rcv have access to the oracle $\mathscr{O}_{\mathsf{GNS}}$.

3.2 Availability

The availability condition of the proposed model requires that the data sent from the sender is delivered to the receiver with high probability. The formal definition is given as follows:

Definition 1 *We say that* $(\mathsf{Gen},\mathsf{Snd},\mathsf{Rcv})$ *satisfies* ε***-correctness*** *if the following inequality holds:*

$$\Pr\left[\begin{array}{c} k \leftarrow \mathsf{Gen}(L,\ell,\varepsilon_{\mathsf{cor}},\varepsilon_{\mathsf{sec}},jit); \\ m \leftarrow \mathsf{Snd}_{\mathsf{C}}(k,s,idx) \end{array} : \mathsf{Rcv}_{\mathsf{C}}(k,idx) = s\right] \geq 1-\varepsilon$$

3.3 Confidentiality and Integrity

Security in the proposed model corresponds to the confidentiality and integrity of transmission data sent through the communication channel. We assume that the adversary in the proposed model possesses read and write access to the global tape and tries to obtain information about the plaintext and/or forge the transmission data. The formal definition of integrity is defined through the following game where the pair of algorithms $\mathscr{A} = (\mathscr{A}_1,\mathscr{A}_2)$ represents the adversary who tries to forge the transmission data under an adaptive chosen message attack.

$\exp(\mathscr{A}^{\mathsf{Gen},\mathsf{Snd},\mathsf{Rcv}})$
1. $k \leftarrow \mathsf{Gen}(L,\ell,\varepsilon_{\mathsf{cor}},\varepsilon_{\mathsf{sec}},jit)$
2. $\mathsf{List}_{\mathsf{Snd}} \leftarrow \emptyset$
3. **for** $i = 1$ **to** L **do**
4. $\quad (s,\mathsf{state}) \leftarrow \mathscr{A}_1^{\mathscr{O}_{\mathsf{GNS}}}(\mathsf{C},\mathsf{state})$
5. $\quad m \leftarrow \mathsf{Snd}_{\mathsf{C}}(k,s,i)$
6. $\quad \mathsf{List}_{\mathsf{Snd}} \leftarrow \mathsf{List}_{\mathsf{Snd}} \cup \{(s,i)\}$
7. $\mathsf{C}' \leftarrow \mathscr{A}_2^{\mathscr{O}_{\mathsf{GNS}}}(\mathsf{C},\mathsf{state})$
8. **for** $i = 1$ **to** L **do**
9. $\quad s \leftarrow \mathsf{Rcv}_{\mathsf{C}}(k,i)$
10. $\quad$ **if** $s \neq \perp$ **and** $(s,i) \notin \mathsf{List}_{\mathsf{Snd}}$ **then return** 1
11. **return** 0

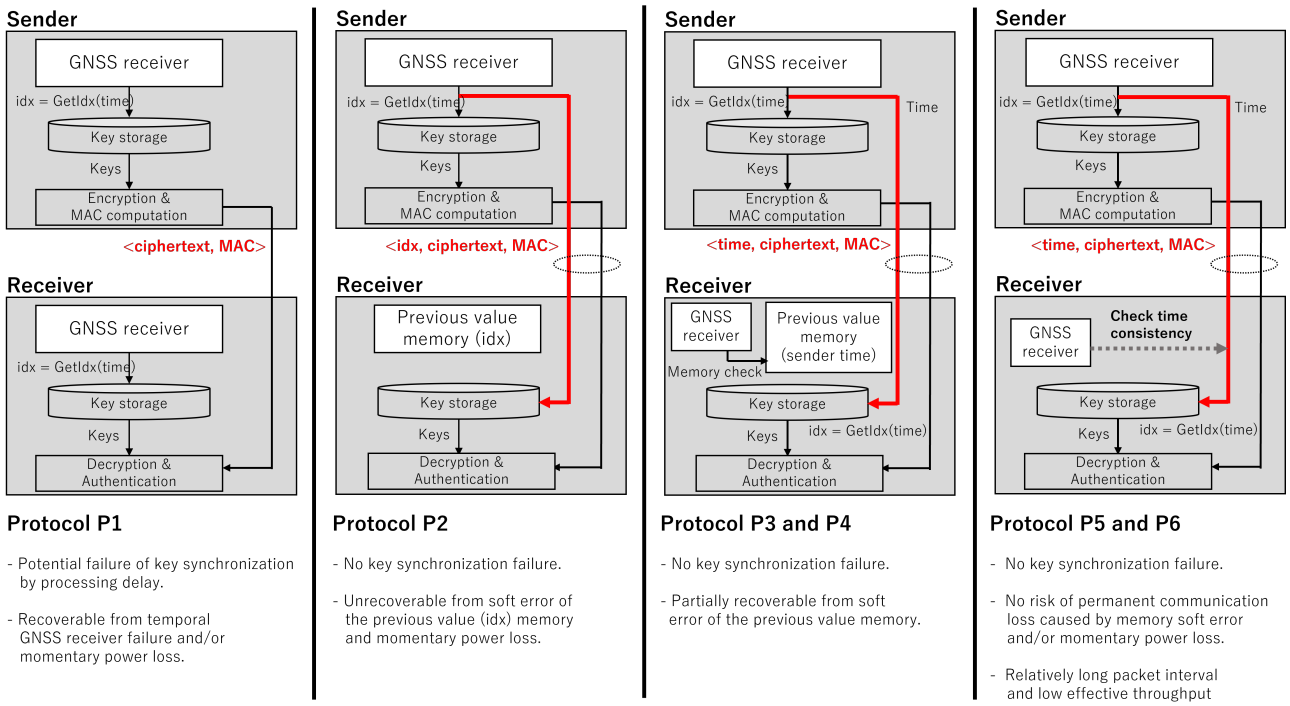

Figure 2 – Our previous protocols P1 $\sim$ P6 [1,2,5]

Definition 2 *We say that* $(\mathrm{Gen}, \mathrm{Snd}, \mathrm{Rcv})$ *satisfies* ε***-unforgeability*** *if the following inequality holds for any (not necessarily polynomial bounded) adversary* $\mathscr{A}$*.*

$$\Pr[\exp(\mathscr{A}^{\mathrm{Gen},\mathrm{Snd},\mathrm{Rcv}}) = 1] \leq \varepsilon$$

Confidentiality requirement in the proposed model is based on perfect secrecy which is first presented by Shannon [3].

Definition 3 *Let* S_i *and* M_i $(i = 1, \ldots, L)$ *be random variables derived from the* i*-th plaintext and the* i*-th transmission data, respectively. We say that* $(\mathrm{Gen}, \mathrm{Snd}, \mathrm{Rcv})$ *satisfies* ***perfect secrecy*** *if it satisfies the following equality for any* $m_i \in \mathscr{M}$ *and for any* $s_i \in \mathscr{S}$ $(i = 1, \ldots, L)$ *where the probability is taken over the probability distribution of* $k \in \mathscr{K}$*.*

$$\begin{aligned}&\Pr[S_i = s_i, \ldots, S_L = s_L | M_1 = m_1, \ldots, M_L = m_L] \\ &= \Pr[S_i = s_i, \ldots, S_L = s_L]\end{aligned}$$

4. Our Previous Protocols P1–P6

In this section, we review six secure wireless communication protocols between an SLV/SS (small launch vehicle/small satellite) and a GS (ground station) [1, 2, 5], which follow the security model described in the previous section.

4.1 Basic Strategies for Protocol Design

All the proposed protocols (Figure 2) are designed to avoid deadlock even in unstable communication environments since the loss of communication is critical in wireless communication between SLV/SS and GS. In designing protocols, we employ matured techniques for encryption and message authentication [3, 12], whose security has been verified by the cryptographic community for a long time. Key synchronization (i.e., guaranteeing that the small satellite and the control base use the same key for encryption and message authentication) is critical to avoid deadlock, and we have proposed several techniques to guarantee key synchronization.

We should note that we assume that both the SLV/SS and GS are equipped with GNSS receivers in all the proposed protocols for achieving key synchronization. We should also note that we use the

words “sender” and “receiver” in explaining the proposed protocols. The sender denotes the entity that will send the transmission data (e.g., GS becomes the sender in uplink communication, whereas SLV/SS becomes the sender in downlink communication). The receiver denotes the entity that will receive the transmission data.
In the first protocol we named P1, both the sender and the receiver receive the time information from the positioning satellite independently and determine the key for encryption and message authentication based on the received data. On the other hand, in the protocol P2, only the sender receives the time information from the positioning satellite to determine the key. The information (which we call index), which enables the receiver to determine the key, is sent by the sender as part of the communication payload. Such protocol allows the attacker to designate the key used in the past communication, which enables the attacker to make the receiver accept the transmission data sent by the sender sometimes before (so-called “replay attack”). To prevent such an attack, the receiver records the value of the last index sent from the sender. It verifies that the key used in the current transmission data is derived after receiving the last transmission data.
The protocols P3 and P4 are improved versions of P1 and P2.[2] Namely, they offer better tolerance against device failure and more robust security against replay attacks, as well as mitigating the security loss caused by a successful attack that might occur with a tiny probability. Please note that there is an essential difference in the packet content between them and P2. As shown in Figure 2, key index value idx (storage address) is sent to the receiver on a P2 packet while the sender’s GNSS time is sent on P3 and P4.

The protocols P5 and P6 are more reliable version of P3 and P4.[3] They incorporate a replay attack detection algorithm using GNSS time information and the use of a state variable ("previous value memory" in Figure 2 is eliminated [1,2]. As a result, in P5 and P6, there is no risk of permanent communication loss following some H/W anomalies, such as soft errors and momentary power shutdown.

4.2 Cryptographic Primitives Employed in the Protocols

Our proposed protocol employs an unconditionally secure authentication code (a.k.a. A-code) to achieve unconditional unforgeability. We give the model of A-code and a construction that we employed in the experimental flight.

A-code (Unconditionally Secure Authentication Code): A-code consists of the following 4-touple $(\mathscr{K},\mathscr{S},\mathscr{M},A)$.

$\mathscr{K}$: The set of shared key
$\mathscr{S}$: The set of source states (plaintext to be sent)
$\mathscr{M}$: The set of transmission data
A : Encoding Function

In A-code, the sender and receiver share the key $k \in \mathscr{K}$. In sending a source state $s \in \mathscr{S}$, the sender encodes s into the transmission data $m \in \mathscr{S}$ by computing $m = (s, A(k,s))$ and sends m through the communication channel. On receiving the transmission data $m' = (s', a')$, the receiver checks whether $a' = A(k,s')$ holds and accepts m' as authentic if the equality holds or rejects m' otherwise.
The security of A-code is defined by the following game where $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ in the game represents an adversary who tries to forge transmission data. Here, we assume that the adversary has access to oracles $\mathscr{O}_{AE}$ and $\mathscr{O}_E$ where $\mathscr{O}_{AE}$ takes s as input and outputs $(s, A(k,s))$, and $\mathscr{O}_E$ outputs $(s, A(k,s))$ for s chosen according to the probability distribution of $\mathscr{S}$. The adversary $\mathscr{A}$ successfully forges a transmission data if $\exp(\mathscr{A}^{(\mathscr{S},\mathscr{M},\mathscr{K},\mathscr{E},\mathscr{D})})$ returns 1 where $\hat{\mathscr{M}}$ appears in the winning condition (i.e., line 3) represents the set of transmission data which oracle outputs during the game.

[2]The difference between P3 and P4 is the addressing method of key storage access and is a relatively minor issue. Please see [5] for further details.

[3]The algorithm of replay attack detection is different between them, and that of P6 is more accurate. We explain the details of P6 in this paper.

$\exp(\mathscr{A}^{(\mathscr{S},\mathscr{M},\mathscr{K},\mathscr{E},\mathscr{D})})$:
1. $\mathsf{state} \leftarrow \mathscr{A}_1^{\mathscr{O}_{AE},\mathscr{O}_E}(\mathscr{S},\mathscr{M},\mathscr{K},\mathscr{E},\mathscr{D})$
2. $(s',a') \leftarrow \mathscr{A}_2(\mathsf{state})$
3. **if** $(s',a') \notin \hat{\mathscr{M}}$ **and** $a' = A(k,s')$ **return** 1
4. **else return** 0

Definition 4 *We say that A-code $(\mathscr{K},\mathscr{S},\mathscr{M},A)$ satisfies ε-**unforgeability** if no adversary $\mathscr{A}$ wins the game with probability better than ε.*

Construction of A-code: Here, we explain an efficient A-code [12] based on a polynomial over a finite field. In this A-code, the key $k = (k_0,k_1)$ is a pair of random elements of $GF(2^n)$, and the source state $s = (s_1,\ldots,s_\ell)$ is an element of $GF(2^n)^\ell$. The encoding function $A(k,s)$ is defined as follows:

$$A((k_0,k_1),(s_1,...,s_\ell) = k_0 + \sum_{i=1}^{\ell} s_i \cdot k_1^{\ell-i+1} \tag{1}$$

We should note that the secrecy of the source state is not guaranteed with the A-code. Therefore, we employ a one-time pad to achieve perfect secrecy. More precisely, we adopt an encrypt-then-mac methodology to achieve unforgeability and perfect secrecy (i.e., first encrypt a plaintext and then apply A-code to the resulting ciphertext).

4.3 Detailed Description of P1 and P2

Here, we explain protocols P1 and P2. The notable feature of protocols P1 and P2 is the use of A-code not only for message authentication but also for extracting payload from the communication stream, which is difficult to distinguish from random noise. We should note that, though we use two different communication channels for uplink C_U and downlink C_D in the actual flight, we do not distinguish which channel we use and use the notation C to denote the communication channel, which is reasonable since the protocol is identical whichever the communication channel is.
We describe the protocol P2 in detail and its difference from P1. As noted, the protocol uses the time information received from the positioning satellite to determine the key used for encryption and authentication among keys stored in the key storage. To make it possible to determine the key, we assume the existence of the algorithm $\mathsf{GetIdx}: \mathscr{T} \to \{0,1\}^T$ where GetIdx takes the time received from positioning satellite with inputs and outputs the index idx which uniquely determines the key stored in the key storage. In protocol P2, idx is sent by the sender as part of the payload, whereas idx is not sent by the receiver in protocol P1 but is independently computed by the receiver upon receiving the payload. [4]

Key Generation: The key generation phase takes the total number of transmission data L sent/received in the lifetime, the bit length of the transmission data ℓ, the security parameters $\varepsilon_{\mathsf{cor}}$ and $\varepsilon_{\mathsf{sec}}$ with respect to correctness and unforgeability, respectively, and parameter jit_{GNS} concerning jitter as input, and generate L keys $(k_{A,1},\ldots,k_{A,L}) \in \mathscr{K}^L$ of A-code $(\mathscr{K},\{0,1\}^{\ell+T},\{0,1\}^{\ell+T+n},A)$, L keys $(k_{S,1},\ldots,k_{S,L})$ of one-time pad. The L pair of keys $k = (k_1,\ldots,k_L)$ where $k_i = (k_{S,i},k_{A,i})$ is stored in key storages of the control base and the small satellite, respectively.
Here, we employ the A-code $(\mathscr{K},\{0,1\}^{\ell+T},\{0,1\}^{\ell+T+n},A)$ satisfying $\min(\frac{\varepsilon_{\mathsf{sec}}}{L\cdot\mathsf{MaxTrial}},\frac{\varepsilon_{\mathsf{cor}}}{L\cdot\mathsf{MaxTrial}})$-unforgeability where $\mathsf{MaxTrial}$ is a parameter of Rcv algorithm to be explained later.

Data Transmission: In sending the plaintext s, the algorithm Snd first computes the index $idx \leftarrow \mathsf{GetIdx}(t)$ based on the time information t obtained via oracle query to $\mathscr{O}_{\mathsf{GNS}}$. Then, it determines the key k_{idx} used for encryption and message authentication. The algorithm then encrypts and authenticates the plaintext using the Encrypt-then-MAC methodology and outputs the resulting data to the communication channel C. The detailed description of the Snd algorithm is as follows.

[4] We leave the detailed implementation of GetIdx out of the scope of this paper.

$\mathsf{Snd}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}(k,s)$
1. $t \leftarrow \mathscr{O}_{\mathsf{GNS}}$
2. $idx \leftarrow \mathsf{GetIdx}(t)$
3. $(k_S, k_A) \leftarrow k_{idx}$
4. $c \leftarrow s + k_S$ // Encrypt s using one-time pad
5. $a \leftarrow A(k_A, (idx, c))$ // Generate authentication tag for (idx, c)
6. **output** $((idx, c), a)$

Data Reception: In receiving the transmission data, the algorithm Rcv must first extract what the sender intends to send from the noisy and jittered channel. To realize data extraction, Rcv extracts interim transmission data $m = ((idx, c), a)$ with appropriate length from the channel C, and determines interim key $k_{idx} = (k_S, k_A)$ (whereas k_{idx} is derived via GNSS query in the protocol P1). Then Rcv checks whether $a = A(k_A, (idx, c))$ holds where A is an function to generate authentication tag. If the equality holds, then Rcv decides that m is correct transmission data and decrypts the ciphertext c by $m = c - k_S$. If this is not the case, Rcv tries the next interim transmission data (that is, Rcv extracts 1-bit shifted data of m with the same length from the channel C). Such a trial is repeated until the correct transmission data is extracted (or the number of trials exceeds the parameter MaxTrial). With this procedure, Rcv can extract the correct transmission data with high probability (i.e., unless the jitter of the communication channel is not so high). The detailed description of the Rcv algorithm is as follows. Here, the variable o appearing in the pseudo-code records the last offset position from which the interim transmission data is extracted, and the variable $prev_{idx}$ records the last index with which the received transmission data was authenticated.

$\mathsf{Rcv}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}(k)$
1. $o \leftarrow o + \ell + n$
2. $cnt \leftarrow 0$
3. **while (**$cnt <$ MaxTrial**) do**
4. $\quad ((idx, c), a) \leftarrow \mathsf{C}[o : o + \ell + n]$
5. $\quad$ **if** $(idx > prev_{idx})$
6. $\quad\quad (k_S, k_A) \leftarrow k_{idx}$
7. $\quad\quad$ **if** $a = A(k_A, c)$
8. $\quad\quad\quad prev_{idx} \leftarrow idx$
9. $\quad\quad\quad$ **output** $c - k_S$
10. $\quad o \leftarrow o + 1$
11. $\quad cnt \leftarrow cnt + 1$
12. **output** $\perp$

As long as the jitter of the communication channel is not so high that Rcv algorithm fails to extract correct transmission data within MaxTrial trials, we can guarantee the correctness of the protocol by invoking Rcv algorithm with appropriate frequency and timing adapting to frequency and timing with which Snd algorithm sends data. Furthermore, as summarized by the following theorem, the security of the proposed protocol is proven regardless of the jitter.

Theorem 1 *The protocol P1* $(\mathsf{Gen}, \mathsf{Snd}, \mathsf{Rcv})$ *satisfies* $\varepsilon_{\mathsf{cor}}$*-correctness,* $\varepsilon_{\mathsf{sec}}$*-unforgeability, and perfect secrecy.*

4.4 Issues with Protocols P1 and P2

Here, we summarize several issues with protocols P1 and P2.
The protocol P1 has such a drawback that key synchronization fails if the gap between the sender's and the receiver's time information is large. This issue is critical since, as we report in [5], a potential gap exists. On the other hand, P1 has the merit that a successful attack causes no further security loss since key synchronization is performed independently for each data transmission.

P2 possesses such a merit that the gap between time information received by the sender and the receiver does not cause failure in key synchronization since only the sender receives time information from the positioning satellite. However, there is a potential risk in P2 that the receiver will fail to accept the legitimate transmission data sent by the sender over a long period if the attacker successfully makes the receiver accept forged transmission data. This undesired situation can happen if the attacker successfully makes the receiver accept forged transmission data $((idx', c'), a')$ where idx' points to an address that will be used in the future. In this case, $prev_{idx}$ variable of Rcv algorithm is set to idx', rejecting the packet until the index value becomes larger than $prev_{idx}$. We should note that such kind a phenomenon can happen not only because of a successful attack but also because of distortion of $prev_{idx}$ value due to temporary power shutdown or soft error caused by high-power radiation ray or even momentary power shutdown of the sender may cause the same phenomenon. We present protocols P3 through P6 to resolve the issues mentioned above.

4.5 Overview of Protocols P3 and P4

In the protocol P2, Snd algorithm sends the key index idx as the part of the transmission data $((idx, c), a)$, and Rcv algorithm accepts idx as authentic if A-code a attached is accepted. In the protocols P3 and P4, we slightly modified P2 in a way that (1) Snd algorithm sends a transmission data of the form $((t^{(S)}, c), a)$ instead of $((idx, c), a)$ where $t^{(S)}$ is the time information received from positioning satellite, (2) Rcv algorithm also receives the time information $t^{(R)}$ from positioning satellite, and rejects the transmission data $((t'^{(S)}, c), a)$ if a condition "$t'^{(S)} > prev_t \ AND \ \delta^{\downarrow} \leq t^{(R)} - t'^{(S)} \leq \delta^{\uparrow}$" does not hold.

Here, $\delta^{\downarrow}$ is a lower bound of the GNSS time difference (= $t^{(R)} - t^{(S)}$) and $\delta^{\uparrow}$ is an upper bound of that. They are constant values and are determined before flight. We will show actual values of $\delta^{\downarrow}$ and $\delta^{\uparrow}$ later in Section 5.

In P3 and P4, it is possible to fix the distorted $prev_t$ variable as follows. Suppose $prev_t > t^{(R)} - \delta^{\uparrow}$ holds. In this case, since the $prev_t$ value is set to the wrong (future) value, we can fix the $prev_t$ value to $t^{(R)} - \delta^{\uparrow}$, which mitigates the security loss caused by successful attack or device disorder due to soft error and temporary power shutdown. We also note that P3 can be seen as an improved version of P1. The difference between P1 and P3 is that P1 uses time information $t^{(R)}$ for directly deriving idx whereas P3 does not use $t^{(R)}$ to derive idx but uses it for verifying the value of index sent by the sender, which resolve the issue concerning the failure of key synchronization due to the gap of time information between the sender and the receiver.

4.6 Overview of Protocols P5 and P6

In P5 and P6, the constant values $\delta^{\downarrow}$ and $\delta^{\uparrow}$ are more precisely determined than in P3 and P4. The condition for replay attack detection is simplified to "$\delta^{\downarrow} \leq t^{(R)} - t'^{(S)} \leq \delta^{\uparrow}$". The use of a state variable $prev_t$ is eliminated, and as a result, P5 and P6 have less risk of permanent communication loss due to receiver malfunctions. In addition, there is no possibility of losing key synchronization, unlike P1, because the same sender time $t^{(S)}$ is used in the sender and receiver for deriving key index value.

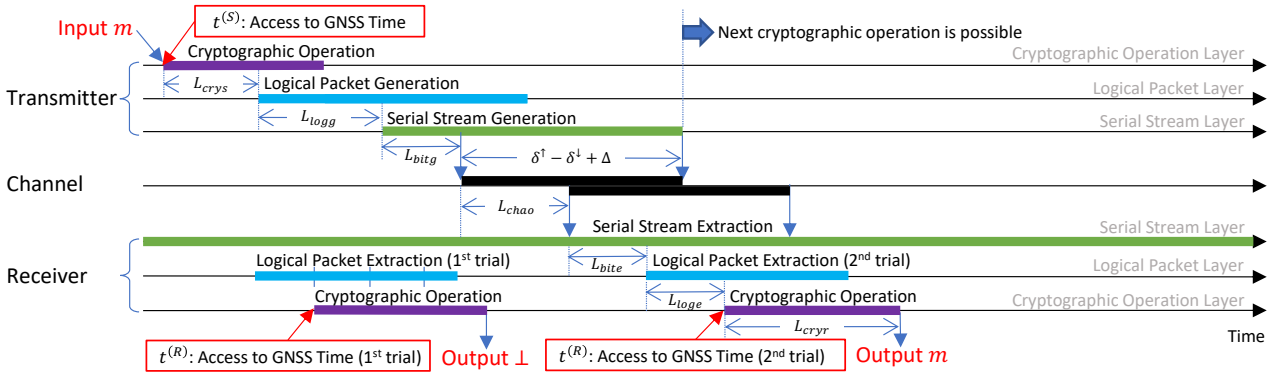

Figure 3 – Flow diagram of the proposed protocols P1−P6

Table 1 – List of theoretical factors affecting time difference [1]

	Factor	Type	Notation	Maximum value	Minimum value
Sender side	GNSS time	Error	$\varepsilon_{\mathrm{S_{time}}}$	${\varepsilon^{\uparrow}}_{\mathrm{S_{time}}}$	${\varepsilon^{\downarrow}}_{\mathrm{S_{time}}}$
	Cryptographic operation	Latency	L_{crys}	${L^{\uparrow}}_{\mathrm{crys}}$	${L^{\downarrow}}_{\mathrm{crys}}$
	Logical packet generation	Latency	L_{logg}	${L^{\uparrow}}_{\mathrm{logg}}$	${L^{\downarrow}}_{\mathrm{logg}}$
	Serial-stream generation	Latency	L_{bitg}	${L^{\uparrow}}_{\mathrm{bitg}}$	${L^{\downarrow}}_{\mathrm{bitg}}$
Receiver side	Channel-out	Latency	L_{chao}	${L^{\uparrow}}_{\mathrm{chao}}$	${L^{\downarrow}}_{\mathrm{chao}}$
	Serial-stream extraction	Latency	L_{bite}	${L^{\uparrow}}_{\mathrm{bite}}$	${L^{\downarrow}}_{\mathrm{bite}}$
	Logical packet extraction	Latency	L_{loge}	${L^{\uparrow}}_{\mathrm{loge}}$	${L^{\downarrow}}_{\mathrm{loge}}$
	GNSS time	Error	$\varepsilon_{\mathrm{R_{time}}}$	${\varepsilon^{\uparrow}}_{\mathrm{R_{time}}}$	${\varepsilon^{\downarrow}}_{\mathrm{R_{time}}}$

To determine the values $\delta^{\downarrow}$ and $\delta^{\uparrow}$, we have conducted a detailed timing analysis of the proposed protocols in [1] by enumerating theoretical factors that affect the time difference between $t^{(S)}$ and $t^{(R)}$. As shown in Figure 3, the packet generation, transmission, and receive operations are executed in parallel (if implemented as hardware) or in a serial (software) manner. In the figure, time slots corresponding to each factor are indicated by the double-headed arrow.
Here, $\delta \equiv t^{(R)} - t^{(S)}$ is estimated as follows, and the constants $\delta^{\uparrow}$ and $\delta^{\downarrow}$ are defined as the maximum and minimum values of δ;

$$\delta = \varepsilon_{\mathrm{S_{time}}} + L_{\mathrm{crys}} + L_{\mathrm{logg}} + L_{\mathrm{bitg}} + L_{\mathrm{chao}} + L_{\mathrm{bite}} + L_{\mathrm{loge}} + \varepsilon_{\mathrm{R_{time}}},$$

where the notation of each term is shown in Table 1. In summary, the sender and receiver algorithm of P6 is described as follows.

$\mathsf{Snd}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}(k,s)$
1. $t^{(S)} \leftarrow \mathscr{O}_{\mathsf{GNS}}$
2. $idx \leftarrow \mathsf{GetIdx}(t^{(S)})$
3. $(k_S, k_A) \leftarrow k_{idx}$
4. $c \leftarrow s + k_S$ // Encrypt s using one-time pad
5. $a \leftarrow A(k_A, (t^{(S)}, c))$ // Generate authentication tag for $(t^{(S)}, c)$
6. **output** $((t^{(S)}, c), a)$ // Minimum send latency $\delta^{\uparrow} - \delta^{\downarrow} + \Delta$ is required (Δ is an arbitrary positive value)

$\mathsf{Rcv}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}(k)$
1. $o \leftarrow o + \ell + n$
2. $cnt \leftarrow 0$
3. **while (**$cnt <$ MaxTrial**) do**
4. $((t'^{(S)}), c), a) \leftarrow \mathsf{C}[o : o + \ell + n]$
5. $t^{(R)} \leftarrow \mathscr{O}_{\mathsf{GNS}}$
6. **if** $(\delta^{\downarrow} \leq t^{(R)} - t'^{(S)} \leq \delta^{\uparrow})$
7. $idx \leftarrow \mathsf{GetIdx}(t'^{(S)})$
8. $(k_S, k_A) \leftarrow k_{idx}$
9. **if** $a = A(k_A, c)$
10. **output** $c - k_S$
11. $o \leftarrow o + 1$
12. $cnt \leftarrow cnt + 1$
13. **output** $\perp$

Please note that in line 6 of the sender algorithm, a throughput requirement to packet transmission is added [1]. The purpose of this requirement is to have a sufficiently large channel-occupation time so that no replay packet can be sent successfully in an absolute time window from $t^{(S)}$ to $t^{(R)}$. As a result, the protocol's maximum throughput and minimum packet interval (if packets are sent periodically) have been limited, while the protocols P1 through P4 don't have such limitations.

5. A Newly Proposed Protocol P7

5.1 A Performance Issue in P5 and P6

We perform an actual performance analysis of P5 and P6 and estimate that maximum throughput is limited to approximately 100kbps on our launcher ZERO [8]. We consider that the result is not specific to our launchers, and the same performance is expected if the protocol is used on any LEO/SSO launchers. The obtained performance result is sufficient for flight command transmission but insufficient for telemetry (vehicle status and other measurements) transmission.

The estimated values of timing parameters in Table 1 are shown in Table 2. A primary factor that limits the total performance is a channel latency L_{chao}. The fluctuation of L_{chao} is significantly significant compared to that of the other factors,[5] because the sight distance between the spacecraft and the ground station changes from nearly 0 to approximately 3000km during flight. Following this parameter estimation, the minimum packet interval is limited to $\delta^{\uparrow} - \delta^{\downarrow} + \Delta \coloneqq L^{\uparrow}{}_{\mathrm{chao}} - L^{\downarrow}{}_{\mathrm{chao}} \coloneqq 10^{-2}$ seconds (corresponds to 100Hz) and also, maximum effective throughput becomes 100kbps assuming the length of a packet is 1000 bits[6] including ciphertext, A-code, GNSS-time information, and the other necessary overheads.

Table 2 – Estimated timing parameter values in our launchers (unit of time: seconds)

Sender	GNSS error	$\varepsilon^{\uparrow}{}_{\mathrm{S_{time}}} = 10^{-5}$	$\varepsilon^{\downarrow}{}_{\mathrm{S_{time}}} = -10^{-5}$
	Cryptographic op.	$L^{\uparrow}{}_{\mathrm{crys}} = 10^{-4}$	$L^{\downarrow}{}_{\mathrm{crys}} = 10^{-8}$
	Logical packet gen.	$L^{\uparrow}{}_{\mathrm{logg}} = 10^{-5}$	$L^{\downarrow}{}_{\mathrm{logg}} = 10^{-8}$
	Serial stream gen.	$L^{\uparrow}{}_{\mathrm{bitg}} = 4 \times T_{chao}$	$L^{\downarrow}{}_{\mathrm{bitg}} = 4 \times T_{chao}$
Channel (sounding rocket MOMO [6])		$L^{\uparrow}{}_{\mathrm{chao}} = 4 \times 10^{-4}$	$L^{\downarrow}{}_{\mathrm{chao}} = 10^{-7}$
Channel (satellite launcher ZERO [8])		$L^{\uparrow}{}_{\mathrm{chao}} = 10^{-2}$	$L^{\downarrow}{}_{\mathrm{chao}} = 10^{-7}$
Receiver	Serial stream extr.	$L^{\uparrow}{}_{\mathrm{bite}} = 10^{-4}$	$L^{\downarrow}{}_{\mathrm{bite}} = 10^{-4}$
	Logical packet extr.	$L^{\uparrow}{}_{\mathrm{loge}} = 328 \times T_{chao}$	$L^{\downarrow}{}_{\mathrm{loge}} = 328 \times T_{chao}$
	GNSS error	$\varepsilon^{\uparrow}{}_{\mathrm{R_{time}}} = 10^{-5}$	$\varepsilon^{\downarrow}{}_{\mathrm{R_{time}}} = -10^{-5}$
	Cryptographic op.	$L^{\uparrow}{}_{\mathrm{cryr}} = 10^{-4}$	$L^{\downarrow}{}_{\mathrm{cryr}} = 10^{-8}$

5.2 Algorithm of P7

To improve the protocol's throughput, we changed the condition of replay attack detection. Our key idea is to minimize the fluctuation of δ by measuring packet propagation delay L_{chao} dynamically in flight and then eliminating L_{chao} from timing factors that determine $\delta^{\uparrow}$ and $\delta^{\downarrow}$. The delay L_{chao} can be measured in flight if sender location $p^{(S)}$ (on the WGS84 coordinate system) and receiver location $p^{(R)}$ are obtained from corresponding GNSS receivers.

A block diagram of the modified protocol P7 is shown in Figure 4, and the algorithm is described below:

$\mathsf{Snd}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}(k,s)$

1. $[t^{(S)}, p^{(S)}] \leftarrow \mathscr{O}_{\mathsf{GNS}}$
2. $idx \leftarrow \mathsf{GetIdx}(t^{(S)})$
3. $(k_S, k_A) \leftarrow k_{idx}$
4. $c \leftarrow s + k_S$ // Encrypt s using one-time pad
5. $a \leftarrow A(k_A, ([t^{(S)}, p^{(S)}], c))$ // Generate authentication tag for $([t^{(S)}, p^{(S)}], c)$
6. **output** $(([t^{(S)}, p^{(S)}], c), a)$ // Minimum send latency $\delta'^{\uparrow} - \delta'^{\downarrow} + \Delta$ is required (Δ is an arbitrary positive value)

[5] In Table 2, the values Ł_{bitg} and Ł_{loge} are defined by a one-bit transmission time T_{chao}. The value of T_{chao} is determined so that the condition $\delta^{\uparrow} - \delta^{\downarrow} + \Delta > 0$ has a solution and can take any value depending on packet length. Because little fluctuation (difference of maximum value and minimum value) is expected, we can ignore Ł_{bitg} and Ł_{loge} from the discussion of the limiting factor of throughput.

[6] In Table 2, we assume the total packet length is 328 bits (in estimating $L^{\uparrow}{}_{\mathrm{loge}}$ and $L^{\downarrow}{}_{\mathrm{loge}}$) based on our actual vehicle design.

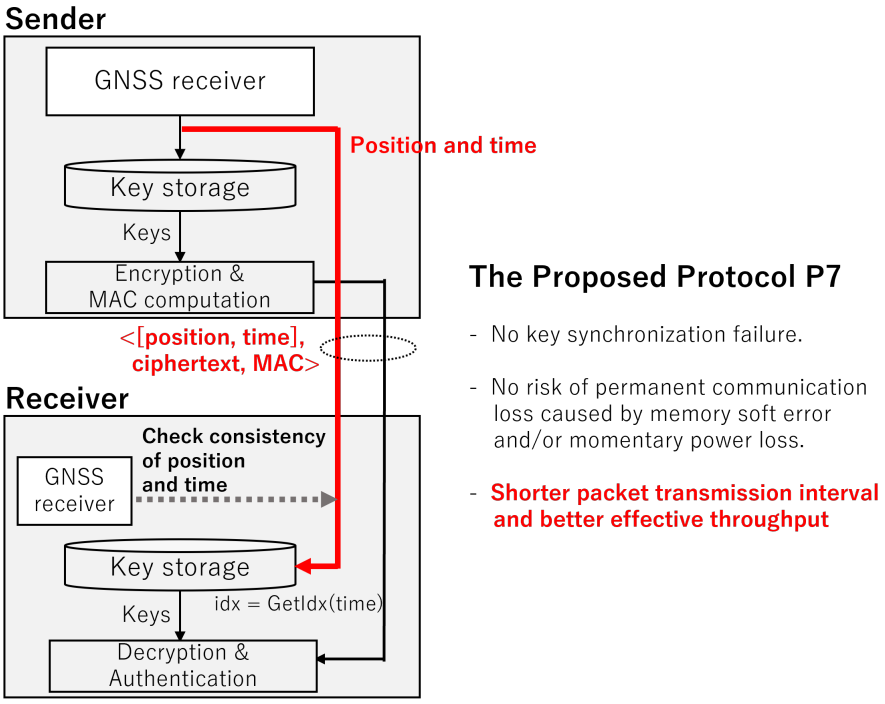

Figure 4 – The proposed fast protocol P7

$\mathsf{Rcv}_{\mathsf{C}}^{\mathscr{O}_{\mathsf{GNS}}}(k)$
1. $o \leftarrow o+\ell+n$
2. $cnt \leftarrow 0$
3. **while** ($cnt <$ MaxTrial) **do**
4. $(([t'^{(S)}), p'^{(S)}], c), a) \leftarrow \mathsf{C}[o : o+\ell+n]$
5. $[t^{(R)}, p^{(R)}] \leftarrow \mathscr{O}_{\mathsf{GNS}}$
6. $pdly \leftarrow \mathsf{GetDelayFromPosition}(p^{(R)}, p'^{(S)})$
7. **if** $(\delta'^{\downarrow} \leq t^{(R)} - t'^{(S)} - pdly \leq \delta'^{\uparrow})$
8. $idx \leftarrow \mathsf{GetIdx}(t'^{(S)})$
9. $(k_S, k_A) \leftarrow k_{idx}$
10. **if** $a = A(k_A, c)$
11. **output** $c - k_S$
12. $o \leftarrow o+1$
13. $cnt \leftarrow cnt+1$
14. **output** $\perp$

The main differences with the algorithm of P6 are that

- a sender position $p^{(S)}$ is included in the packet (line 6 of the sender algorithm),
- L_{chao} is computed using a function $\mathsf{GetDelayFromPosition}(p1, p2)$ (line 6 of the receiver algorithm), and
- the judgement of replay attack is done using δ' instead of δ, which is defined by

$$\delta' = \varepsilon_{\mathrm{S_{time}}} + L_{\mathrm{crys}} + L_{\mathrm{logg}} + L_{\mathrm{bitg}} + \varepsilon_{poserror} + L_{\mathrm{bite}} + L_{\mathrm{loge}} + \varepsilon_{\mathrm{R_{time}}},$$

  where $\varepsilon_{poserror}$ is a measurement error of propagation delay L_{chao}. The constants $\delta'^{\uparrow}$ and $\delta'^{\downarrow}$ are defined as the maximum and minimum values of δ', as well.

5.3 Evaluation of Performance Improvement in P7

Like P5 and P6, the throughput of P7 is determined by the fluctuation of δ', and it mostly depends on the value of $\varepsilon_{poserror}$. It is considered that the actual value of $\varepsilon_{poserror}$ is affected by the following factors:

GNSS measurement error on orbit: GNSS positioning is often used in LEO satellites, GN&C systems, and AFTS (Autonomous Flight Termination System) of launch vehicles. The accuracy of positioning on orbit can be better than $1-10$ meters [13]. Even if there is an unlikely error of 30km, the corresponding measurement error of propagation delay is only 10^{-4} seconds, much smaller than the fluctuation of L_{chao}.

GNSS measurement error near the ground: It is well-known that the accuracy of GNSS positioning near the ground worsens because of the multipath effect caused by ground launch facilities, etc. However, we can easily avoid the effect by disabling the propagation delay measurement at the very early phase of the launcher's ascent.

GNSS module output latency: Conventional low-cost (yet space-grade) GNSS modules generate output at the rate of $1-10$Hz, i.e., there is a $0.1-1$ seconds latency between the position measurement and its output. This latency causes a $0.78-7.8$ km difference from the actual position at the orbital speed, yet the corresponding difference of propagation delay is 3×10^{-5} seconds at most.

The overall estimation of the fluctuation of $\varepsilon_{poserror}$ ($\fallingdotseq$ that of δ') is 10^{-5} to 10^{-4} seconds, and therefore approximately 10kHz packet interval and 10Mbps throughput (if the packet length is 1000 bits) are achievable.
As a result, as shown in Figure 2, Figure 4 and Table 3, the new protocol P7 resolves all potential risks in P1$-$P6, i.e., key synchronization loss, communication loss by state variable destruction, and low throughput.

Table 3 – Summary of differences between the protocols P1$-$7

	P1	P3$-$P4	P5$-$P6	P7
Key index value	Sender and receiver derives independently using their GNSS.	Sender specifies (the index is written in packet)	Sender specifies (derived from send-time $t'^{(S)}$ in packet)	Same as P5$-$P6
Method of finding a replay attack	———- (none)	A state variable $prev$ in receiver	Comparison of send-time $t'^{(S)}$ in packet and receive-time $t'^{(R)}$	Added a dynamic delay measurement based on GNSS positioning to P5$-$P6
Theoretical throughput limitation	None	None	Approximately 10kbps	Approximately 10Mbps

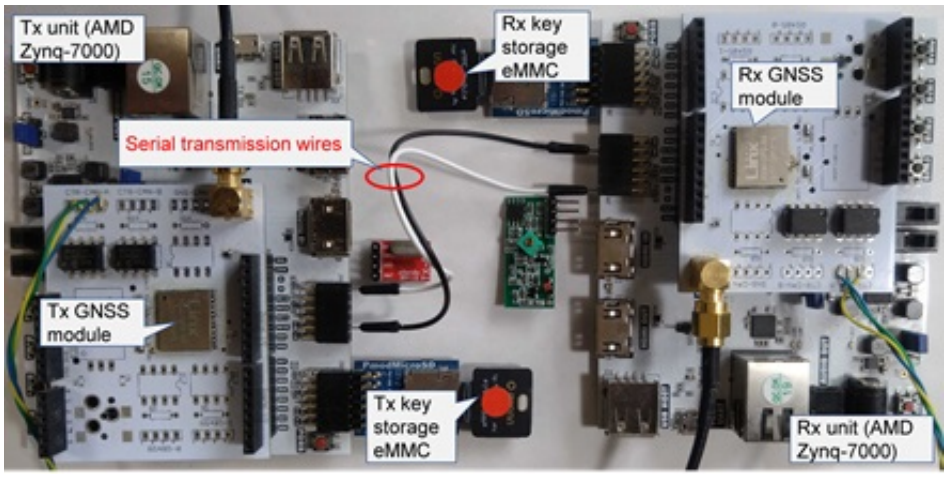

Figure 5 – Prototype boards

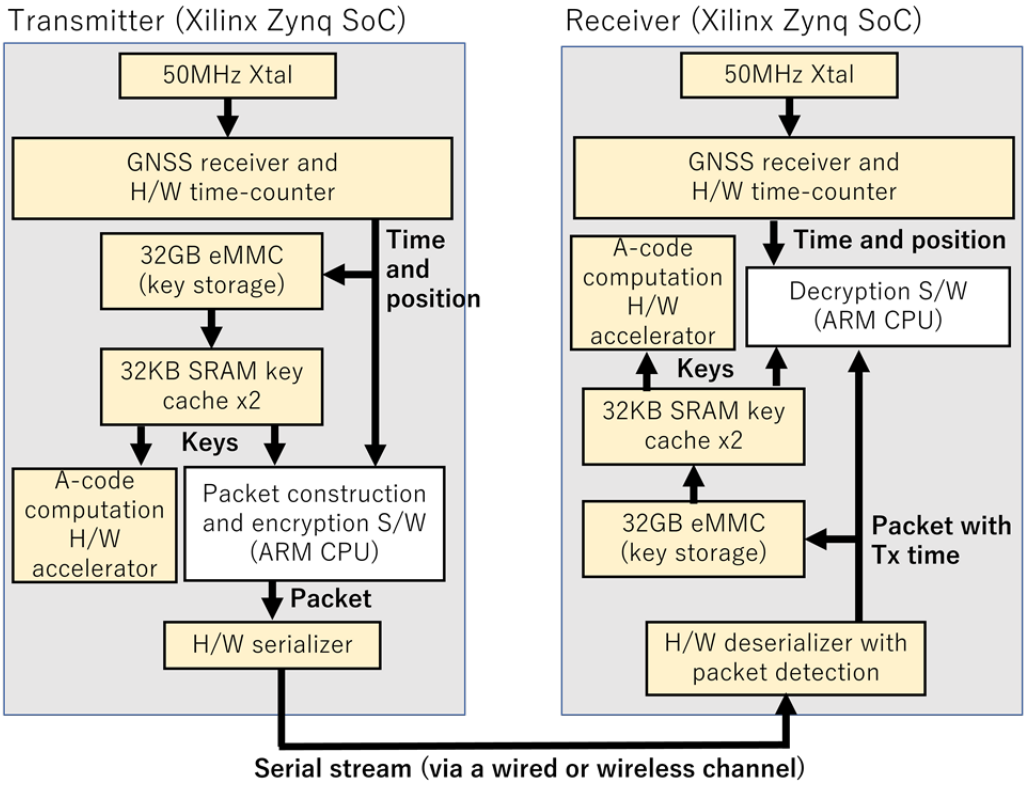

Figure 6 – Dataflow diagram for P7 performance evaluation

5.4 Prototype Implementation of P7

We made a prototype implementation of P7 to confirm the operation performance. The testbed and its block diagram (yellow boxes correspond to H/W and white boxes correspond to S/W) are shown in Figure 5 and Figure 6, respectively. The following H/W components are used in the testbed:

Computation Device: An AMD Zynq-7020 chip with two ARM Cortex-A9 cores operating at 650MHz clock speed and an FPGA block is used as an implementation platform. The same device has been used in MOMO [7] and will also be adopted in ZERO [8].

CPU: The ARM core executes encryption, decryption, and data transfer control.

H/W IPs: The time counter, (de)serializer, and the other interface peripherals are implemented as dedicated H/W IP cores on the FPGA block. A-code computation is executed on a high-speed H/W accelerator at approximately 12.8Gbps data rate. (A 128-bit data block is processed in one clock cycle at 100MHz.)

Key storage: One commercial-grade 32GB eMMC memory device is used as key storage. Unlike the previous flight demonstrations on MOMO, using an SD card is rejected since a contact failure can happen in the socket connector. During this test, we evaluated multiple eMMC devices made by several chip makers, but little difference was found regarding device performance.

GNSS receiver: We use the same module as the previous demonstrations on MOMO. However, our satellite launcher ZERO will use a custom-made GNSS receiver unit.

Physical layer: We do not use actual flight-rated high-power radio transmitter and receiver modules in this experiment environment because an official license is required in Japan. Instead, we connect the sender and receiver by a short-length wire. A simple asynchronous data transmission protocol (similar to conventional UART) is used on the wired link.

On this testbed, we have confirmed that packet transfer is possible at approximately 10.2Mbps throughput without any authentication or decryption errors. The performance achieved by P7 security

operations is sufficient to simultaneously handle single or multiple high-resolution video streams and other telemetry.

6. Conclusion

In this paper, we have improved the throughput of our proposed wireless communication protocol to an order of 10Mbps by incorporating the use of GNSS positioning. The protocol provides the highest security level, called information theoretic security, which cannot be broken even if an attacker has unlimited computing power. No risk of key synchronization loss is guaranteed theoretically, and the use of state variables is minimized to improve resilience against temporal H/W failures.
Possible future work is twofold: (i) confirm the accuracy of propagation delay measurement by GNSS positioning, using a flight-rated high-power radio transmitter that can be used for long-distance communication, and (ii) constructing a backup algorithm against an accidental loss of GNSS position measurement.

References

[1] Maki Yoshida, Sumio Morioka, and Satoshi Obana, “Secure Communication via GNSS-based Key Synchronization,” Proc. of Work-in-Progress in Hardware and Software for Location Computation (WIPHAL 2023)

[2] S. Morioka, S. Obana, and M. Yoshida, “A Highly Reliable Key Synchronization Framework in Information Theoretically Secure Wireless Communication for Small Spacecrafts,” Proc. of the 34th International Symposium on Space Technology and Science (ISTS), Session J-3, 2023-j-12, June 2023.

[3] C. E. Shannon: Communication Theory of Secrecy Systems, Bell System Technical Journal, Vol.28-4, pp.656–715, Oct. 1949.

[4] https://public.ccsds.org/default.aspx

[5] S. Morioka, S. Obana, and M. Yoshida, “Flight Demonstration Results of Information Theoretically Secure Wireless Communication on a Sounding Rocket MOMO,” Proc. of the 33rd International Symposium on Space Technology and Science (ISTS), Session J-2, 2022-j-06, Feb. 2022.

[6] Interstellar Technologies Inc., https://www.istellartech.com/launch/momo (Last accessed on 26th Oct. 2023).

[7] S. Morioka and T. Inagawa, “Design Requirements and Implementation of Avionics System on Commercial Space Launch Vehicles,” THE JOURNAL OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS Vol.105 No.4 pp.(1)-(8) April 2022, https://www.journal.ieice.org/bin/pdf_link.php?fname=k105_4_275e (Last accessed on 28th Feb. 2023).

[8] Interstellar Technologies Inc., https://www.istellartech.com/launch/zero (Last accessed on 26th Oct. 2023).

[9] C.E. Shannon, “A Mathematical Theory of Communication,” Bell System Technical Journal, vol.27, no.3, p.381, 1948.

[10] C.E. Shannon and W. Weaver, The Mathematical Theory of Communication, University of Illinois Press. ISBN 978-0-252-72546-3, 1998.

[11] M. Bellare and C. Namprempre, “Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm,” Journal of Cryptology, vol. 21, no. 4, pp. 469-–491, 2008.

[12] B. den Boer: A Simple and Key-Economical Unconditional Authentication Scheme, Journal of Computer Security, Vol.2, pp. 65–71, 1993.

[13] O. Montenbruck, F. Kunzi and A. Hauschild, "Performance assessment of GNSS-based real-time navigation for the Sentinel-6 spacecraft," GPS Solutions, Vol.26, No.12 (2022). https://doi.org/10.1007/s10291-021-01198-9

Copyright Statement