|  |  |
| --- | --- |
| label_document |  |

|  |
| --- |
| Executive Summary Report  *Main Outcomes and Findings of the Testbed for Bundle Protocol Security (BPSEC) project under ESA Contract No. 4000137376/22/D/AH* |





|  |  |  |  |
| --- | --- | --- | --- |
| Title: “Executive Summary Report” | | | |
| Issue: 1 | | Revision: 0 | |
| Authors: Alessandro Cammarano, Jorge Finochietto, Juan José Grosso | | Date: 17/11/2023 | |
| Approved by: Juan José Grosso | | Date: 17/11/2023 | |
|  |  |  |

© COPYRIGHT EUROPEAN SPACE AGENCY, 2023

The copyright of this document is vested in the European Space Agency. This document may only be reproduced in whole or in part, stored in a retrieval system, transmitted in any form, or by any means electronic, mechanical, photocopying, or otherwise, with the prior permission of the Agency.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reason for change | | Issue | | Revision | | **Date** | |
|  | First version of the document. |  | 1 |  | 0 |  | 17/11/2023 |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Issue 1 | | Revision 0 | | | | | |
| **Reason for change** | | **Date** | | **Pages** | | **Paragraph(s)** | |
|  | First version of the document |  | 17/11/2023 |  | All |  | All |

Table of contents:

1 Introduction 4

2 Project objectives and WORK DONE to reach them 4

3 DEVELOPED System 6

4 Problems Encountered during the project 7

5 MAIN OUTCOMES AND FINDINGS 9

6 Future work 11

# Introduction

This executive summary report is written as part of the ESA project “Testbed for Bundle Protocol Security (BPSEC)” under ESA Contract No. 4000137376/22/D/AH coordinated by Osmium (re-branded from RDI network to Osmium by the time of this deliverable and used throughout this text) and with GTD Sistemas de Información (GTD) and Fundació i2CAT (i2CAT) as sub-contractors.

# Project objectives and WORK DONE to reach them

The overall objective of the project was to implement and validate, on a virtualised testbed, the latest Bundle Protocol (version 7) and the Bundle Protocol Security specification as specified in the relevant standards (RFC9172 and RFC9173), addressing also operational concerns and challenges from multiple perspectives including security. As part of the project, two realistic proof of concept applications (Lunar and EO communications) have been validated over the testbed allowing both to verify the correctness of the implementation of the testbed itself and to identify additional issues and critical aspects of the BP/BPSEC protocol specifications. Any identified finding has been analysed, formalised and shared with the standardization workgroups.

To reach this overall objective, there were a number of sub-objectives that are itemized in the next table, with the key work done to reach them.

|  |  |
| --- | --- |
| **Objective** | **Work Done** |
| Analysis of existing IETF standards, identification of problem areas and proposition of solutions  WPs involved: WP1 | Osmium conducted the review of previous versions of the standards and related projects and liaised with international experts, such as NASA-JPL's ION team, for additional feedback.  In total, five documents were thoroughly reviewed (i.e., RFC 9171, RFC 9172, RFC 9173, IEEE SCC21, Bundle-in-Bundle Encapsulation).  By understanding the current state-of-the-art, the team identified and systematically documented sensitive and problematic aspects of both BP and BPSEC. These aspects had an impact on the project and drove the design and implementation of the BPSEC protocol for ESA. This assessment served as an input to the scope of implementation on subsequent WPs, together with potential solutions to address these issues (see section 3 for a list of the main problems and solutions). |
| Definition of the scope of the BP (version 7) and the relevant BP security specification implementation  WPs involved: WP1, WP2, WP6 | Osmium defined the scope of the Testbed for BPSEC and enumerated details on the requirements that served as a basis for the subsequent WP3-WP4 (in charge of GTD).  The activity identified and analysed a set of applicable threats to DTN networks based on BP. Based on them, different threat scenarios have been investigated, i.e., five test scenarios with the attacks, effects, mitigations, and policies, where applicable. This was done considering the threat scenarios, cases and the specificities of the application scenarios (i.e., Earth Observation Science Data Downlink and Lunar Communications). |
| Derivation of associated test cases scenarios and definition of the testbed system requirements  WPs involved: WP2, WP6 | As part of WP2, considering the previous objectives and the specific two test scenarios, this activity provided and explained a comprehensive set of 28 requirements corresponding to three different specifications, namely, Test Scenario, Test Case and Test Report. |
| Implementation of the IEFT BP v7 and the relevant BP security specification  WPs involved: WP4 | GTD followed a formal process (ECSS) for the implementation of the BPSEC specification.  Stemming from TN1 and TN2, GTD defined the software requirements for the implementation of BP & BPSEC and ESA’s BPv7 implementation was reused as starting point for the development of BP Security protocol.  The design of the BP security protocol implementation was done using the Unified Modelling Language and the coding in Java, to be in line with ESA’s BPv7 implementation.  A verification plan was defined for the verification of the implemented BPSEC and foresaw a complete set of unit tests implemented using the JUnit library and managed with TM4J. The unit tests building, and execution was automated using Jenkins.  The verification results were included in SVR2 with all tests passed. |
| Design, implementation and validation of the testbed architecture  WPs involved: WP3 | Common Open Research Emulator (CORE) was selected as the base of the testbed development as the best option considering scalability, configurability, interoperability, and observability aspects. Performance and automation were also considered as test scenarios needed to be run by the test bench on a standard computer in a reasonable amount of time and without user intervention.  The design of the testbed was done using the Unified Modelling Language and the language was the same used by the chosen network simulator.  GTD designed the SW for maintainability from the beginning and conducted iterative development and regular reviews that helped to improve the quality of the developed SW.  A software verification and validation plan was defined for the verification and validation of the testbed and foresaw a complete set of functional and operational tests implemented using a standard scripting language (e.g. bash, python, etc.) and managed with TM4J.  The execution of validation tests was fully automated and automatically generated result reports containing the validation tests pass/fail results and detailed output for analysis purposes. |
| Definition and implementation of two proof of concept applications  WPs involved: WP6 | i2CAT described the configuration and implementation of the PoC scenarios using an auxiliary software for PoC scenarios generation (SW3) (that has been implemented on purpose).  WP6 studied in depth the Earth Observation Science Data Downlink Scenario (EOSDDS) and the Lunar Communications Scenario (LCS) and provided a summary of the studied missions, described the architecture and models followed in this type of scenarios and derived a Proof-of-Concept scenario for the testbed and BPSEC validation. Furthermore, a security architecture study was done in compatibility of the required security architectures and threats for the PoC scenarios. |
| Validation of the BP and BP security specification  WPs involved: WP5 | Osmium conducted the validation process of the BPSEC protocol and provided a detailed report of the execution of test cases.  The validation process was designed using the testbed software, the Wireshark v4.0.1 BPSEC dissector, and the ION software v4.1.2. The target of the validation process is the BPSEC implementation; the Testbed provided a user interface to exploit its capabilities, running simulations of a communications network, while the BPSEC dissector enabled examining the results of these simulations.  The validation process was based on a series of test cases developed from the threat analysis and test scenarios defined in WP2 and each test scenario specified a configuration of communication nodes, possible threats that apply to that configuration, and one or more security policies designed to prevent those threats, when possible. For each scenario, multiple test cases were proposed to demonstrate that the security policies can be implemented using the functionalities and expressiveness provided by the protocol implementation so that the data communication can be carried out effectively and in a secure manner.  The validation process was iterative, running all the test cases over a specific release of mentioned software implementations.  During this activity, several interactions with NASA’s JPL team were needed as the interoperability tests with ION were rather complex and, in the end, not fully possible due to restrictions on ION’s ITAR-free version and due to bugs and reduced set of features (mainly for logs). |
| Findings and lessons learned with recommendations for the standards and future systems implementing them  WPs involved: WP7 | Osmium regularly interacted with ESA and reviewed different documents, apart from those used in WP1. For instance, these were: “DTN Bundle Protocol Security (BPSEC) COSE Context”, “CCSDS 734.5-P-1.1, Bundle Protocol Security Specification. Red Book. Issue 2”, “CCSDS 350.9-G-2, Cryptographic Algorithms (Green Book) and CCSDS 354.0-R-2, Symmetric Key Management (Red Book, Issue 2) .  For these, Osmium documented in technical notes or formally via email the specific comments that were considered as lessons learnt and recommendations.  Also, during the several meetings held, Osmium shared its points of view on the different topics discussed with ESA and within this document, a summary of the recommendations and future work proposed to address the main problems encountered during the project is proposed in Sections 5 and 6. |

# DEVELOPED System

Figure 1 serves as a comprehensive overview of the various software modules developed for this project's scope. In addition to showcasing these modules, the diagram captures also additional components that have been used as a technical input for its design or for real simulation configuration, e.g., the description of the proof-of-concept scenarios, threat analysis, security policies. At the forefront of this software architecture are the primary modules: CORE (Common Open Research Emulator) which formed the foundational basis for subsequent testbed development (SW1) and the bundle protocol (BP) sourced from ESA extended to host the BPSEC implementation (SW2). Complementing these core elements are SW3, responsible for dynamically configuring the testbed based on predefined PoC scenarios, and SW2\_1, a dedicated WireShark BP/BPSEC dissector instrumental in meticulously analysing captured traffic for insights and evaluation.

SW1 and SW2 collectively establish a comprehensive testbed capable of simulating realistic scenarios akin to lunar communication and earth observations. This platform facilitates the formulation of security policies crucial for safeguarding communication within these scenarios. The simulation logs, encompassing traffic captures, serve as foundational data for scientific analysis. Moreover, they play a pivotal role in evaluating the accuracy and comprehensiveness of network configurations, with a specific emphasis on assessing security policies.

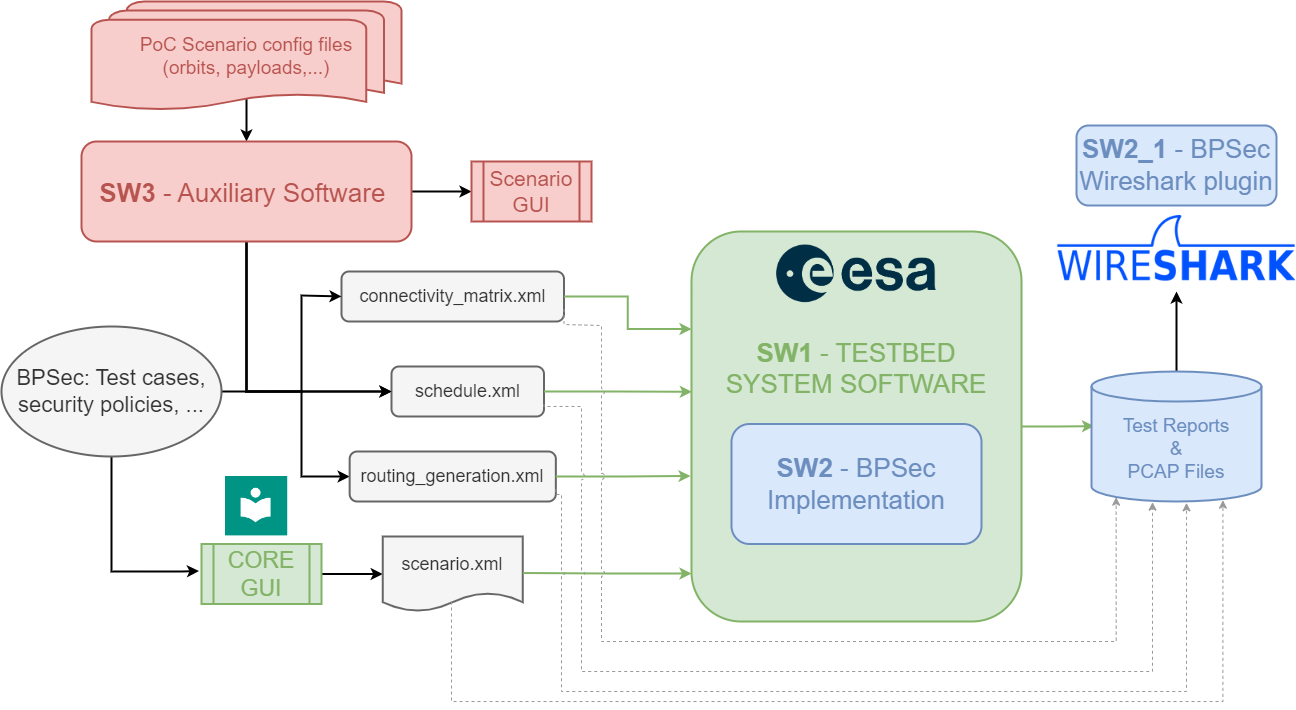


Figure 1: Overall architecture highlighting the implemented modules.

# Problems Encountered during the project

During the project a few issues were found and addressed as follows:

1. *Ambiguity between RFC9172 and RFC9173 when authentication tag is appended to the ciphertext in the block-type-specific data field.*

During the implementation of the confidentiality operation, a problem was observed when using Wireshark to correctly dissect the BCB of a bundle. This occurred because of a potential ambiguity between RFC9172 and RFC9173:

* RFC9172 states that there must be one entry in the Security Results array for each entry in the Security Targets field of the security block.
* RFC9173 states that the BCB-AES-GCM security context produces an authentication tag that can be stored either in the Security Result field or appended to the generated ciphertext. If the authentication tag is included in the ciphertext, the Security Result MUST NOT be included in the BCB for that Security Target.
* Regarding the authentication tag, the decision taken by our implementation is that this tag is added to the ciphertext in the block-type-specific data field of the security target block.

Initially, and following RFC9173, our initial assumption was that it was not necessary to add an entry in the Security Result field, given that the authentication tag was added to the ciphertext. However, based on Wireshark dissection error, we realized that even when the authentication tag is placed in the ciphertext, there still has to be an entry in Security Results (RFC9172), but that entry must be an empty array. This fact is not explicitly stated in RFC9173 and is expected to be clarified in future versions in order to avoid potential interoperability issues between different implementation of the protocol.

1. *Default policy in ION (forward instead of drop)*

Normally, the actions a node takes with a bundle containing (or not) security operations, i.e., blocks, depend on the given node’s security policy (in our implementation this behaviour is achieved by applying in a row the configured security rules). Nevertheless, certain scenarios may prompt default decisions when all applicable security rules have been applied or when no rules are present. Consider, as an example, the following scenario:

1. A node sources a BIB targeting an extension block (e.g., age block).
2. An intermediate node that does not have security rules, modifies the extension block and breaks its integrity.

The question then arises as to whether it would be appropriate for the intermediate node to delete the BIB, given that the integrity of the target is no longer valid. This was precisely the decision taken by our implementation in line with the considerations made in section 8.3, even though this behaviour is not specified nor prohibited in RFC9172.

This led to an interoperability error and tests that did not pass successfully, in cases with ESA BPSEC nodes and ION nodes, since ION nodes do not implement this behaviour and keeps BIB blocks even when the integrity of the target block is compromised.

The final decision was to align the ESA BPSEC implementation with the behaviour of ION in this regard. Nevertheless, such situation should be explicitly addressed by the RFC to avoid different understanding potentially resulting into interoperability problems.

1. *ION without security context from RFC9173*

Initially, most of the tests designed included both ESA BPSEC nodes and ION nodes. The ION documentation states that the security contexts defined in RFC9173 are implemented and functionally operational. However, in the testing stage and after some email exchanges, it became clear that this functionality is provided with the ITAR (International Traffic in Arms Regulations) controlled ION-INB (ION NASA Baseline) only. Therefore, the interoperability tests had to be reduced in their initially intended scope.

1. *Fragments can reach an acceptor before being assembled*

Even if security operations cannot be applied to fragments, it is still possible that an acceptor receives a bundle fragment. In this case, it is important that the assembly process of all fragments is completed before applying the security policy. Otherwise, fragments may be discarded as they arrived if processed by the acceptance function.

In addition, bundle lifetimes must be configured to take into account the additional delays caused by having to wait for all the fragments to be assembled before applying the security operation of verification or acceptance. If the lifetimes are tight, bundles may be discarded.

More, BPSEC might deeply limit the routing paths as the fragments should all reach a verifier/acceptor node to be processed correctly. A fragment passing through a different path might potentially jeopardize the correct reception of the bundle itself at the reception node.

1. *All three extension blocks defined in RFC9171 are designed to work on a hop-by-hop basis, end-to-end policies cannot be proposed.*

All three extension blocks defined in RFC9171, i.e., Previous Node, Bundle Age, Hop Count blocks, require to be modified (updated) at each hop, resulting in mutable data. Hence, integrity or confidentiality services need to be handled also on a hop basis, which limits the proposal of end-to-end security policies.

1. *Limitations on expressing security policies (lessons learnt from our proposal), potentially adding logical expressions*

One of the focuses of this project was to explore the expressive power of security rules to define security policies for the mitigation of the identified security threats (see TN1). Initially, the specification section of security rules did not support the use of wildcards (\*). As a result, it was not possible to manage cases in which either a node A or a node B could have acted as the source of a security operation (BIB or BCB), and therefore the BIB/BCB should be accepted in either case and rejected otherwise.

The conception of more complex scenarios, such as those of the Proof of Concept, highlighted this limitation, which led to the security rules being reviewed and modified to allow wildcards in the Security Source field of the specification.

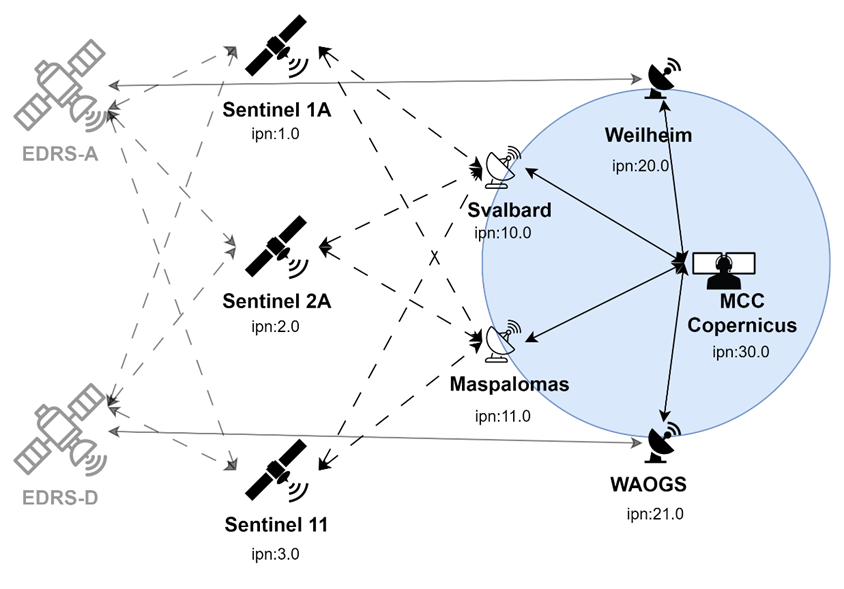
In addition, other possibilities were discussed to allow more expressive rules that might be necessary in even more complex scenarios. In particular, consideration was given to the concept of "meta-rules," consisting of sub-rules, each conditionally executed akin to an "if-elseif-else" structure, usually found in imperative programming languages.

On the other hand, it is clear that the expressive power of rules entails a trade-off in their design. More expressive rules allow the definition of more flexible security policies, but at the same time increase the complexity of their usage/configuration and of their implementations.

# MAIN OUTCOMES AND FINDINGS

A comprehensive analysis of BPSEC for securing space missions was conducted. The primary focus was on assessing the capabilities of BPSEC for implementing security policies required by common threats, including data interception, data manipulation, masquerading, and replay attacks.

This work, among other achievements, introduced a method to address the absence of formalized security policies in BPSEC specification, i.e., RFC 9172. By introducing the concept of security rules - utilizing conditions, specifications, and outcomes - we established a framework to design comprehensive security policies that can be applied to any BPSEC node of the network. These policies might not only ensure block integrity but also extend protection to entire bundles, allowing, at the same time, confidentiality for payload content when necessary. Additionally, we analysed and eventually recommended the use of scope flags to broaden integrity protection, especially for sensitive blocks like primary and payload blocks.



*Figure 2: Earth Observation Science Data Downlink Scenario*

Relying on the available implementation of ESA’s Bundle Protocol (BP) (RFC 9171), this work implemented the Bundle Protocol Security (BPSEC) (RFC 9172) and its default security contexts (RFC 9173) as part of a flexible and scalable testbed. The developed testbed facilitated rigorous validation across varied scenarios like Earth Observation (Figure 2) and Lunar Communication (Figure 3). Insights obtained from these Proof-of-Concept (PoC) scenarios offer critical conclusions on both the BPSEC protocol and its implementation.

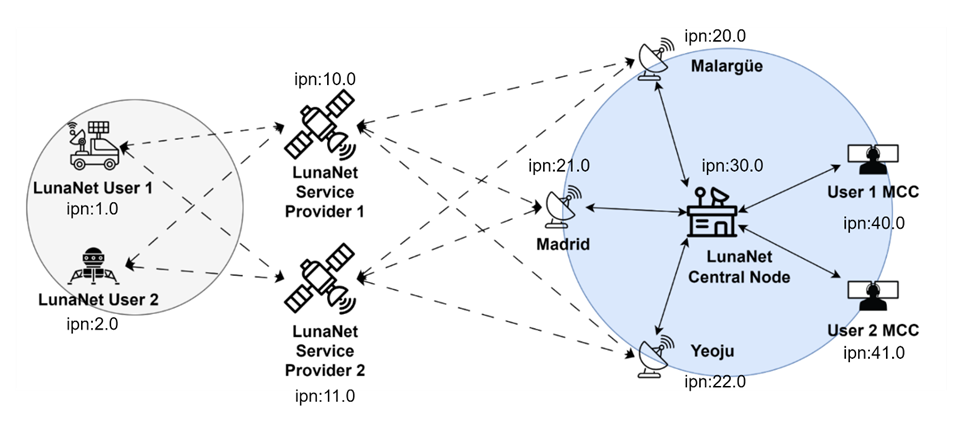


Figure 3: Lunar Communication Scenario

An observed benefit of BPSEC functionality is the capability to assure the bundle security irrelevant of the taken path. An important requirement, for example, in securing the integrity of critical telemetry and telecommands that travel through unpredictable paths and untrusted intermediaries. This also makes the provision of an end-to-end security services for the user data irrelevant to the intermediary nodes’ security capabilities, meaning BP nodes without BPSEC capabilities may be involved in the bundle’s hops. Lunar and interplanetary scenarios may in a future involve multitude of intermediate BP nodes of third parties unrelated to the mission, which thanks to the mentioned benefit, will not require BPSEC in order to participate in the network.

It has also been observed that the BPSEC policy rules implementation can be scalable and suited for future scenarios with large number of nodes and complex networks. The possibility to use wildcards in the rule conditions avoids adding redundant rules and provides the tools to prevent the spread of non-secured bundles by using “catch-all" rules. This allows to accept the operations coming from a set of possible sources, useful when the path a bundle will take is not fully deterministic. On the other side, the number of parameters to specify allow tailoring rules on a per user/endpoint basis, which makes it possible to provide diverse security services throughout the network without interference between rules.

# Future work

As a future work we foresee the following open challenges:

1. Improve the expressiveness of the security policy definition by combining and condition “sub-rules” that can lead to better control on the policies in scenarios were routing is dynamic.
2. Analyse the performance of executing security policies to optimise its resource consumption, potentially including break rules, or ordering its execution base on some criteria.
3. Key management needs also to be investigated to better understand its distribution and lifecycle, considering the impact when a node is compromised, and its keys could be accessed by a malicious agent. Due to the nature of DTN networks, the time to update or renew keys may be significant and leave the network at risk for a long period.
4. Network topology and its routing dynamics may pose challenges in deploying properly security policies at nodes. On the one hand, considering that a bundle may traverse any node in the network impose a deployment of security policies for all potential traffic flows on each node while also requiring enough keys to act as verifier is needed. This has a strong impact on the number of keys each nodes needs to keep stored, introducing a significant risk in case the node is compromised. On the other hand, a more limited routing or analysis of feasible paths, could limit the security policies to only traffic flows that may pass through the node and reduce the required keys. This illustrates that there is a trade-off on the routing dynamics and the security policies.
5. Scenarios with multiple service providers introduce security policies that can be either local to each domain or multi-domain, hence, requiring different distributions of keys and introducing different risks if a node is compromised. A potential solution to keep keys inside each domain is to consider safe passageways between providers limiting key sharing to only those nodes interconnecting these domains.