Information Security in Time- and Space-Partitioned Architectures for Aerospace Systems*

João Carraca, Ricardo C. Pinto, João Craveiro and José Rufino

University of Lisbon, Faculty of Sciences, LaSIGE
Campo Grande, 1749-016 Lisboa, Portugal
{jcarraca,rcp,jcraveiro}@lasige.di.fc.ul.pt, jmrufino@ciencias.ulisboa.pt

Abstract

Time- and Space-Partitioned systems are a current trend in aerospace systems and in autonomous vehicles in general. Such systems employ a partitioned environment through separation of applications in logical containers called partitions. Time and Space Partitioning (TSP) ensures that partitions do not mutually interfere in terms of fulfilment of real-time and addressing space encapsulation requirements. In this paper we present an architecture for future TSP systems and its extension of concerns into the security domain. We will describe the security components that make this architecture well-suitable for the construction of systems with Multiple Independent Levels of Safety and Security (MILS).

Keywords: Time and Space Partitioning, Interpartition Communication, Information Security, Multiple Independent Levels of Safety and Security

1 Introduction

Is it possible to develop a computer-based architecture where multiple applications with a heterogeneous set of attributes (safety and security requirements, criticalities, real-time requirements, origins/developers) can coexist without interfering with each other in terms of safety and security?

The interest of the aerospace industry in finding an answer for this question stems in a large extent from a need for reduced Size, Weight and Power (SWaP)

*This work was partially supported by the EC, through project IST-FP7-STREP-288195 (KARYON) and by FCT, through project PTDC/EEI-SCR/3200/2012 (READAPT), through LaSIGE Strategic Project PEst-OE/EEI/UI0408/2014, and Individual Doctoral Grant SFRH/BD/72005/2010.

consumption. Moreover, there is also a need for mission's development and operational costs savings including certification expenses [15, 13]. On the other hand, in the aerospace domain, information security is a crucial concern, specially in a dual-use scenario, where two or more stakeholders share the same spacecraft platform with different purposes. Stakeholders operate one or more instruments exclusively for their own interests and purposes, wishing to keep confidential not only the instrument results but also the operational scheme. Consequently, the spacecraft platform could be the target of security issues that could have an impact on the spacecraft's safe operation. Future space missions, with their increasing trends for complexity and cooperation between government agencies and commercial enterprises, will certainly require the establishment of security requirements to ensure the authenticity, confidentiality, integrity, and availability of their data [14].

One approach to address all the above requirements, without increased certification costs and providing a complete separation of applications is to use Time and Space Partitioning (TSP). TSP is a concept for safety-critical systems in which applications with mixed criticalities and different requirements (including information security requirements) may coexist in the same computing platform. TSP separates applications into logical containers called partitions, with respect to both time and space domains. Temporal partitioning concerns partitions not interfering with each other's timeliness. Spatial partitioning means that applications running in one partition cannot access any addressing space outside those belonging to that partition. These two properties ensure that faults (accidental/intentional) are contained to their domain of occurrence, preventing them from propagating to other partitions [15, 12].

Motivated by concerns arising both from safety and security perspectives, this paper describes the embedding of security components in TSP architectures, making possible the design and implementation of systems with Multiple Independent Levels of Safety and Security (MILS) [3, 11, 12].

Outline Section 2 describes a TSP architecture for aerospace systems. Section 3 explains information security in a dual-use satellite operational scenario. Sections 4 and 5 describe how security features can be integrated in TSP architectures to support the incorporation of security concerns. Section 6 closes the paper with concluding remarks and future work directions.

2 AIR Architecture for TSP Systems in Space

The **A**RINC 653 **I**n Space **R**eal-time operationg system (AIR) innovation initiative stemmed from the interest of the European Space Agency (ESA) in the adoption of the ARINC 653 [1] concept for space on-board software, and aimed at the utilization of Components Off the Shelf (COTS), namely exploiting the use of the Real-Time Executive for Multiprocessor Systems (RTEMS) [7], a free/opensource Real-Time Operating System (RTOS). However, the AIR activities went further ahead, and resulted, not only in the intended proof of

concept, but also in the definition of a general architecture for an ARINC 653-compliant RTOS, allowing the co-existence of different RTOS kernels in different partitions [10, 9, 4].

AIR implements the architectural principle of robust TSP [9]. The modular design of the AIR architecture, illustrated in Figure 1, relies on the AIR Partition Management Kernel (PMK) to enforce robust TSP. A Partition Operating System (POS) is provided per partition, being foreseen the use of different operating systems among the partitions, either RTOS or generic Non-Real-Time (NRT) ones [4]. Each POS is wrapped by the AIR POS Adaptation Layer (PAL) abstracting its particularities from other AIR components thus ensuring flexibility and independence in the integration of each POS [9, 10].

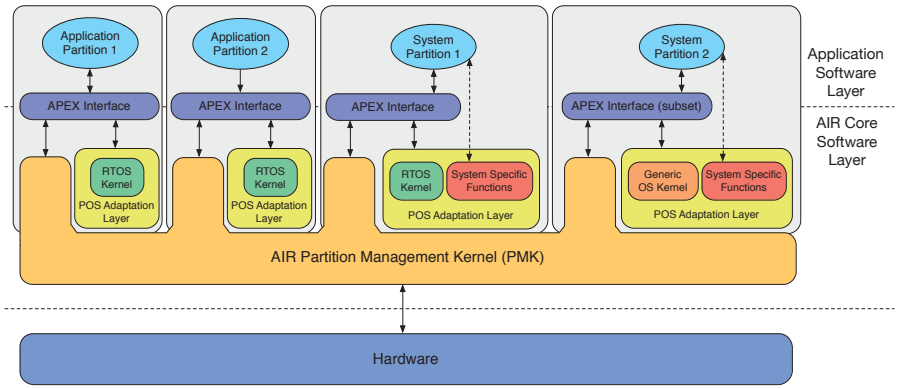

Figure 1: AIR architecture for TSP systems

At the Application Software Layer (Figure 1), applications consist of one or more processes, which make use of the services provided by an Application Executive (APEX) interface, derived from the ARINC 653 specification [1]. In addition, an application hosted in a system partition may bypass the standard APEX interface and invoke system specific functions provided by the POS, as illustrated in Figure 1.

2.1 Partition Management Kernel

The PMK is a component transversal to the whole system (Figure 1) playing a major role in achieving dependability, by ensuring robust TSP. Essentially, the PMK is a simple microkernel that efficiently handles: partition scheduling, partition dispatching and support to interpartition communication [9].

2.1.1 Time Partitioning

Time partitioning, as illustrated in Figure 2, is achieved through a two-level hierarchical scheduling scheme. In the first level, a partition schedule is repeated cyclically over a Major Time Frame (MTF), according to the specified in a

Partition Scheduling Table (PST). A PST is defined offline in configuration files at system integration time [9, 10].

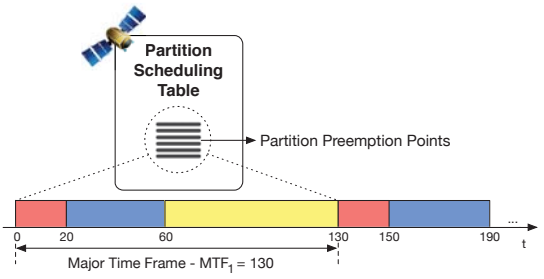

Figure 2: Securing time partitioning

At each clock tick, the PST is checked to test if a partition preemption point has been reached. If that is the case, a switch occurs between the active partition (which currently holds the processing resources) and the heir partition, which will hold the processing resources until the next partition preemption point is reached.

In the second level, processes compete inside each partition for processing resources according to the native process scheduler of each POS [9, 10].

2.1.2 Space Partitioning

Space partitioning ensures that is not possible for an application to access the memory of any other application running on a different partition. AIR follows a highly modular design approach, as illustrated in Figure 3, where spatial partitioning requirements, specified in configuration files at system integration time, are described through a high-level processor-independent abstraction layer, mapped in runtime to processor memory protection mechanisms, such as those provided by a hardware Memory Management Unit (MMU).

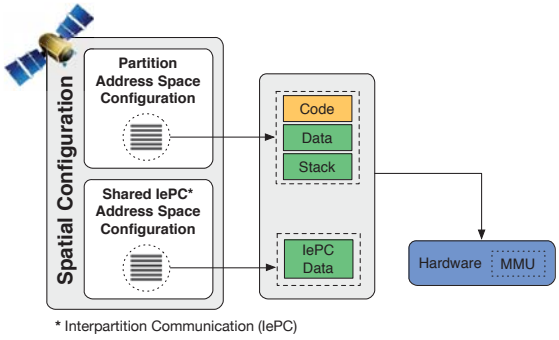

Figure 3: Enforcing space partitioning

A set of descriptors is provided per partition, primarily corresponding to the several levels of execution (e.g., application, operating system and AIR PMK) and to its different memory sections (i.e., code, data and stack). When a partition switch occurs, the mapping into processor specific descriptors needs to be updated in runtime [9, 10, 2].

2.2 Interpartition Communication

Interpartition Communication (IePC) is related with spatial partitioning (Figure 3) in the sense that implies the use of APEX interface services encapsulating and providing the transfer of information from one partition to another without violating spatial partitioning constraints [9, 4]. Interpartition communication is achieved through communication channels configured to operate in sampling or queuing modes, as defined in the ARINC 653 specification [1]. In sampling mode, each new instance of a message overwrites the previous message. The receiving partition always accesses the latest message. In queuing mode, messages are queued and therefore a new instance of a message does not overwrite previous ones [4]. With interpartition communication, messages are transferred from one partition to another through unidirectional channels, with specific access rights, without violating spatial containment restrictions. Provided communication channel endpoints are not located in different physical platforms, a shared memory paradigm is used for the communication channel implementation, as illustrated in Figure 4.

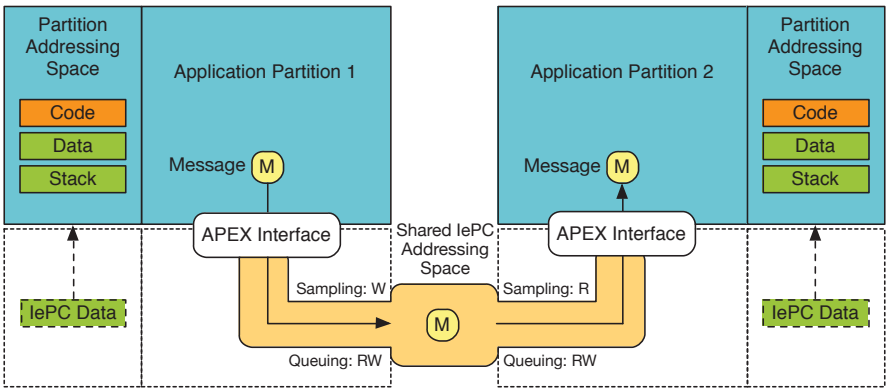

Figure 4: IePC channel

However, only the communicating partitions are allowed to access the commonly assigned shared IePC addressing space. The sender writes the message into the shared IePC addressing space; the receiver performs message reading from the same shared IePC addressing space. The memory protection mechanisms ensure that each partition can only access the addressing space belonging to the partition, along with the shared IePC addressing spaces the partition is allowed to use. In sampling mode the sender has write only (W) permissions and

the receiver read only (R) permissions on the allowed IePC shared addressing space. In queueing mode, the sender has write only (W) and the receiver has read only (R) permissions - from an information flow point of view. However, both read and write (RW) permissions are required - from a queue management point of view. The same does not apply to the sampling mode because there are no management operations. Interpartition communication channels end-points, permissions, maximum message size and maximum number of messages are completely defined at system configuration time (see Figure 3). The memory protection mechanisms guarantee that the communication channels are not accessible by unauthorized partitions, as illustrated in Figure 5.

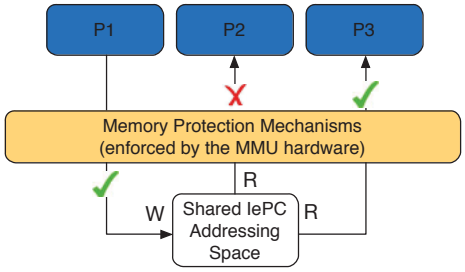

Figure 5: Memory access and protection mechanisms

Partition P1 and P3 are allowed to communicate with each other through an interpartition communication channel. P2 expresses its misbehaviour trying to access the same channel without having access permissions. As we can see, when P2 tries to read the memory addressing space shared by P1 and P3, the memory protection mechanisms hinder P2's intention and an exception will be raised to the appropriate fault handling entities, such as ARINC 653 Health Monitor [2]. In a TSP system, interpartition communication mechanisms are thus secure by design, being this property ensured by the memory protection mechanisms. Thus, it fulfils the security requirements represented in Table 1 usually required for secure communication.

Table 1: Secure Channel Requirements

Confidentiality	Information cannot be read by unauthorized applications
Integrity	Information cannot be modified without detection
Authenticity	Information is sent by trusted applications

The confidentiality property is ensured by TSP memory protection mechanisms. The remaining properties must be secured by additional mechanisms.

3 Information Security in TSP: Use-Case

Information security is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. To better understand the problem of information security in TSP a dual-use satellite scenario will be used, identifying the security requirements associated to the several applications sharing the spacecraft platform. Further we will detail the security components and strategies most adequate to the characteristics of TSP systems and which ones enable the fulfilment of the identified security requirements.

3.1 Use-Case Scenario

Consider the following scenario where two distinct authorities share a satellite platform with the intention to save development, launch and operational costs. Each authority owns an instrument used for civil defence purposes through specific planetary science observations. While spacecraft control operations are performed under the authority of the mission control, each civil defence authority operates its instrument exclusively for its own purposes. On-board instruments are:

- **Payload Fire**: intended for premature detection and prevention of forest fires. This authority wishes to keep the data produced by its instrument secure from the rest of the functions.
- **Payload Water**: detects changes in the global water cycle, loss in snow and ice and monitors the global mean sea level rise caused by climate changes. This authority wishes to keep the data produced by its instrument secure from the rest of the functions.

Figure 6 shows all on-board functions that we are considering for our dual-use spacecraft including the following avionics related functions:

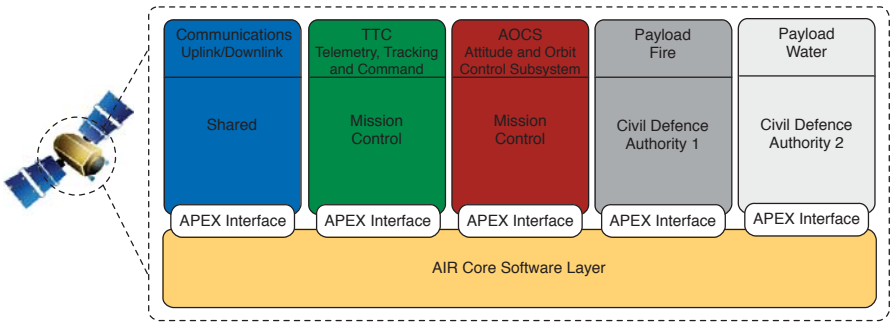

Figure 6: Dual-use spacecraft TSP computing platform

- **Attitude and Orbit Control Subsystem (AOCS)**: provides attitude information and maintains the required spacecraft attitude during all phases of the mission, starting at spacecraft separation from the launch vehicle and throughout its operational lifetime;
- **Telemetry, Tracking and Command (TTC)**: provides control of spacecraft via the ground mission control and informs periodically about the spacecraft position and status;
- **Communications**: provides an uplink/downlink for communication between the spacecraft and the ground segment and transfers the commands issued from the ground mission control to the TTC subsystem. All other spacecraft functions share this application for communication with the ground segment.

Other functions may be hosted in the spacecraft platform, such as Fault Detection, Isolation and Recovery (FDIR) [8]. However, for the sake of simplicity, only a fundamental set was chosen. In a TSP platform, all spacecraft functions are hosted in partitions.

3.2 Communication Requirements

Partitions communicate with each other through interpartition communication channels. The information exchanged can be classified as confidential or non-confidential. Secure communication is needed when two partitions are exchanging confidential information and do not want a third party, which may have access to the communication channel, to be aware of the content. For that they need to communicate in a way not susceptible to eavesdropping or interception. In TSP, any interpartition communication channel is secure by design and therefore may convey both confidential and non-confidential information, with the exception being when information flows share common resources.

The diagram of Figure 7 illustrates the required communication channels between the several spacecraft functions and between the spacecraft platform and the ground segment. Since the Communications partition serves several information flows (partitions Water, Fire and TTC to/from the ground segment) these need to be encrypted if confidentially needs to be secured. Information flows from these three partitions would also be mutually exposed, if not encrypted. Occasionally, some data from Payload Water is sent to AOCS for attitude calibration. Calibration is performed through well known and invariant calibration targets (e.g., a structure on Earth's surface under the control of the Civil Defence Authority 2) captured by the Payload Water observation instrument. To lighten the load of Payload Water, data is sent encrypted simultaneously to the Communications and AOCS, with an extra processing burden at AOCS with respect to decrypting actions.

Outside the TSP platform, both the uplink and the downlink need to be secure because the information exchanged with the ground segment cannot be exposed. Inside the TSP platform, avionic functions are mostly mission-critical

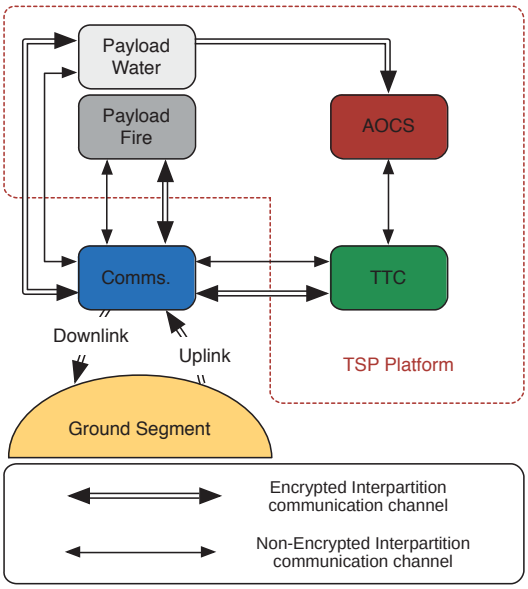

Figure 7: Interpartition communication requirements

(safety) and process non-confidential information (security) while partitions Fire and Water (and occasionally AOCS) process confidential information.

Finally, partitions exchanging both confidential and non-confidential information, need two communication channels. The non-confidential part is transmitted via non-encrypted interpartition communication channels while the confidential part is transmitted through encrypted interpartition communication channels, as illustrated in Figure 7.

4 Information Security Implementation in TSP

Depending on which channel endpoint (if any) encrypts the data, one may have several configuration options for interpartition communication channels.

4.1 Interpartition communication without cryptography

The first case we analyse concerns a secure channel that simply transfers a message (M) of plain data and therefore does not perform any encryption/decryption action neither at the source nor at the destination endpoints. Figure 8 illustrates such configuration. Since the channel is secure, they can be used to convey both confidential and non-confidential information. One example of such a channel is the transfer of ground commands, from TTC to AOCS (see Figures 7 and 8).

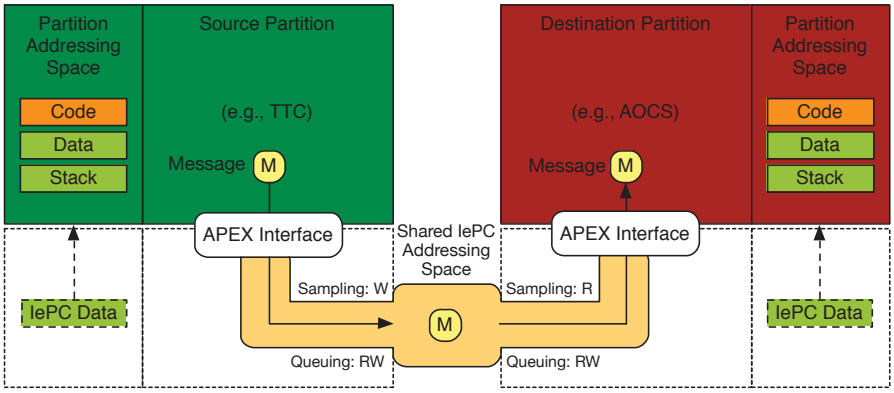

Figure 8: Non-encrypted IePC channel

4.2 Encryption at the TSP Platform

The second configuration option to be analysed concerns message (M) encryption being performed at the endpoint (source) located at the TSP system. The data is not decrypted at the TSP platform, being forwarded (encrypted) to a farther endpoint, for example, an endpoint located at the ground segment. Figure 9 illustrates a communication link from Payload Fire to the ground segment.

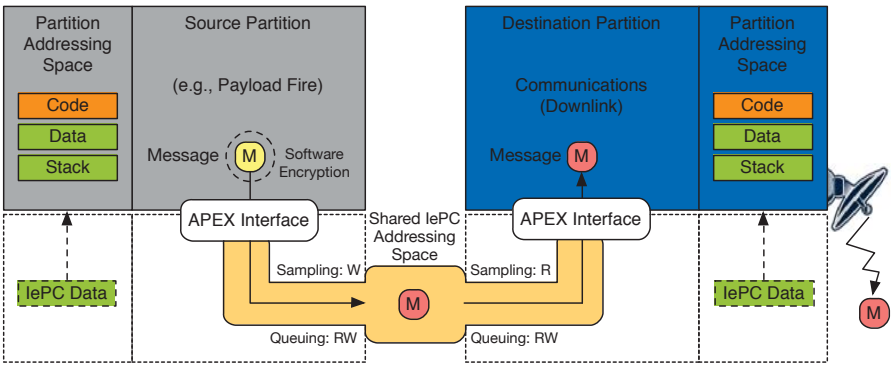

Figure 9: Encrypted IePC channel (encryption at the TSP platform)

In this case, the information flow from Payload Fire leaves the TSP platform through the Communications partition, being transmitted through the downlink. Outside the platform and at the Communications partition there is no protection which means that Payload Fire needs to protect the information before sending it. To that end, Payload Fire encrypts the data (e.g., through software encryption) and sends it to the Communications partition through an interpartition communication channel. The Communications partition forwards the encrypted data that, once received at the ground segment, may be decrypted

and the original information retrieved.

4.3 Decryption at the TSP Platform

The next interpartition communication channel configuration option is dual of the previous one, i.e., the data is encrypted outside the TSP system, for example, at the ground segment. Once the (encrypted) message (M) is received at the Communications partition, it is forwarded by this partition to its final destination at the TSP platform, where the message data is decrypted. Figure 10 illustrates an example, being the final destination Payload Fire. The communication link from the ground segment to the spacecraft platform needs to be protected, together with data handling at the Communications partition.

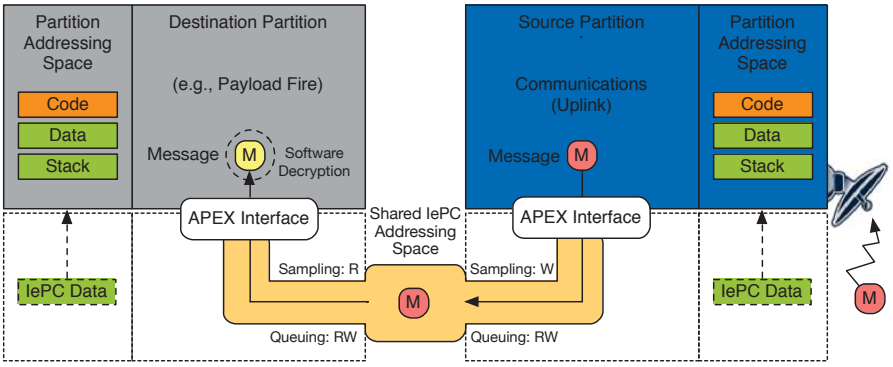

Figure 10: Encrypted IePC channel (decryption at the TSP platform)

The encrypted data is transmitted through the uplink and then transferred to Payload Fire through an interpartition communication channel. Upon reception, Payload Fire decrypts the data and retrieves the original information.

4.4 Encryption and decryption at the TSP Platform

Finally, the last interpartition communication configuration option, specifies that message (M) data is encrypted and decrypted at endpoints located at the TSP platform as specified in Figure 11. This is the case of the Payload Water to AOCS communication channel specified in Figure 7.

Figure 11 illustrates an encrypted channel from Payload Water to AOCS. The information is confidential and only a sub-part is used by AOCS for satellite calibration purposes. In this case, the memory protection mechanisms are not sufficient because part of the information is still confidential to AOCS. In order to secure the information exchanged, Payload Water encrypts (e.g., software encryption) the information and sends it to AOCS. Upon message reception, AOCS decrypts (e.g., software decryption) part of the information and retrieves the calibration information.

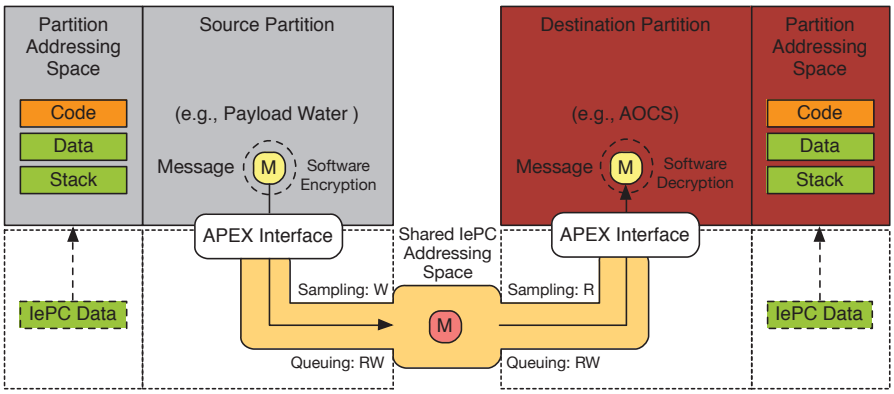

Figure 11: Encrypted IePC channel (encryption and decryption at the TSP platform)

5 AIR Security Component

By enforcing confidentiality, the memory protection mechanisms alone ensure a safe and secure interpartition communication service inside the TSP platform. To enforce the additional properties identified in Table 1 (integrity and authenticity) additional mechanisms are required. In particular, *integrity* and *authenticity* properties require the appendage of Message Authentication Codes (MACs) to the messages encapsulating the data to be exchanged. The provision of these services is achieved by embedding the required security components - {en,de}cryption and MAC - directly in the AIR architecture.

5.1 Architecture

The embedding of the components raises concerns in two fronts: **security**, for the service must be trusted and tamper-proof; **performance**, for it should be able to deliver the service with a reasonable throughput. Both issues can be solved by mapping the security functions into hardware. This approach enables security by using the physical barrier given by the permanent nature of the mechanisms embedded in hardware. It also enables performance, by mapping the (usually) computationally intensive cryptographic algorithms directly into efficient hardware mechanisms, which also offload the processing elements from cryptographic tasks.

The novel AIR Security Component providing cryptography services is enclosed in a logical container named Hardware Security Module (HSM), and is shown in Figure 12, together with its relation with the remainder of the AIR architecture. The already existing hardware platform is also depicted, with a processing element and Input/Output (I/O) interfaces. The HSM entity presented in Figure 12 is comprised of:

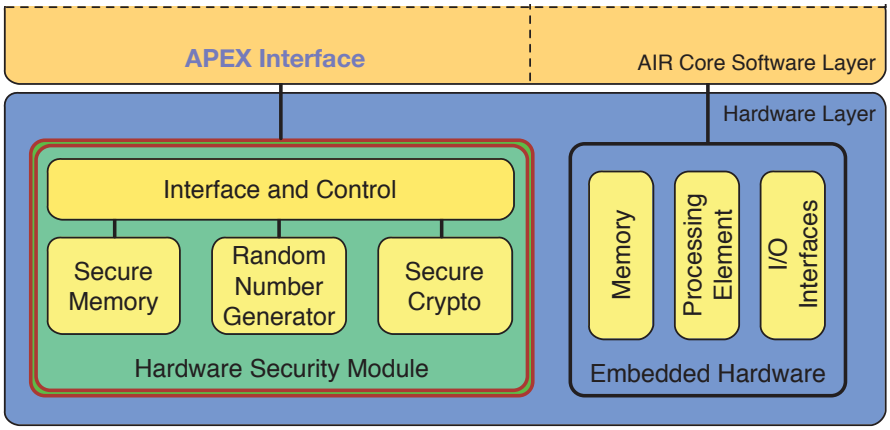

Figure 12: Hardware Security Module Architecture

- **Secure Memory**, a non-volatile data storage containing cryptographic key(s) and sensitive configuration parameters;
- **Random Number Generator**, for key generation purposes;
- **Secure Cryptography**, implementing machinery for cryptographic algorithms used for data encryption and decryption, data integrity enforcement, origin verification, key generation and management;
- **Interface and Control**, providing a control and data interface with the APEX interface software components supporting the interpartition communication services. The data interface comprises input and output resources for data streaming.

Additionally, all components are enclosed inside a physical boundary, which prevents that internal data and processes can be intercepted, copied/cloned, or manipulated yielding to non-authorized use or compromise of internal secrets.

Choosing the cryptographic technologies to be mapped into the Secure Cryptography component of Figure 12 has both performance and key management concerns. Symmetric cryptography is known for delivering better performance over a public/private key scheme, and the usage of a single key for encryption and decryption simplifies the management of the keys. With that in mind the algorithm selected to provide the encryption and decryption services was the Advanced Encryption Scheme (AES) technology [5]. For the cryptographic hash primitive which will create the MACs it was selected the AES Cipher-based MAC (CMAC) algorithm [6].

The HSM is responsible for generating and managing all the communication encryption keys. Since the number of keys increases with the number of partitions/applications, a scalability issue may arise due to an increase in key management and protection needs. In the AIR Security Component this issue

is addressed through the utilization of the *key chain* concept. The key chain is a hierarchical structure, where a Root Key is used to encrypt and decrypt all the other keys. The structure of the key chain is shown in Figure 13.

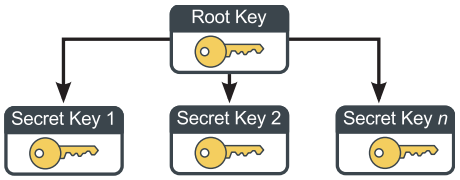

Figure 13: Hierarchical Key Chain

The importance of the Root Key is paramount, for it is the one capable of decrypting all the other keys - and its storage is unencrypted. Therefore, it must be stored in a secure memory, on-chip. All the other keys can be stored in off-chip memory, due to their encrypted nature. This key chain organization enhances key secrecy while reducing on-chip storage needs.

5.2 Operation

Figure 14 illustrates the implementation of a secure interpartition communication channel using the previously described HSM for cryptographic operations. The operation illustrated in Figure 14 is as follows:

- Making use of the APEX interface, the Source Partition sends a request for encryption passing a buffer containing the original data, M_{SP}, a reference to the memory address of the destination buffer, M_{Ch}, and the buffer size. The HSM then performs all crypto-related operations over the original data and outputs the encrypted data directly to the destination buffer, M_{Ch}, via Direct Memory Access (DMA).
- The embedded DMA engine allows the HSM to access system memory independently of the Central Processing Unit (CPU). Through DMA, the CPU (under control of the APEX interface components) initiates the transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller when the operation is done.

Figure 14 shows only the information flow from the Sending point of view. However, upon reception, the Destination Partition passes to the HSM, through the APEX interface, a reference to the communication channel and, implicitly, the address of the buffer containing the encrypted data, M_{Ch}, the memory address of the buffer where the HSM will directly put the decrypted data, M_{DP}, and its size. Then, through a secure interpartition communication service, we can support communication between partitions with different security requirements for the information exchanged, ensuring a tight control of information flow between them.

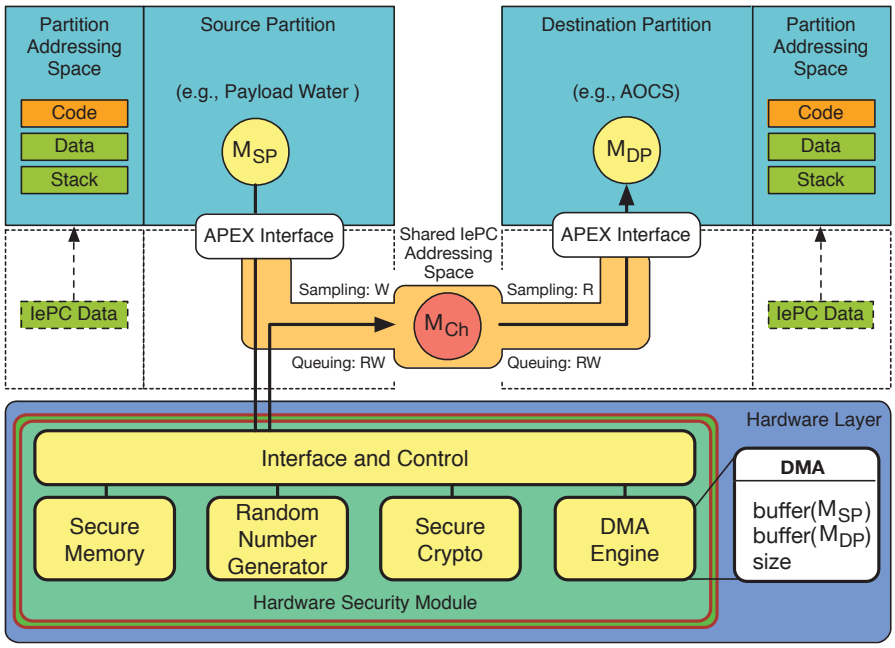

Figure 14: Secure Interpartition Communication

6 Conclusions and Future Work

Time- and Space-Partitioned architectures are the state-of-the-art for the next-generation of aerospace applications, allowing the embedding of the several spacecraft functions in a single computing platform. An application running on a TSP system is given the guarantee that segregation exists, both in the time and space domains, i.e., safety. However, the security properties were and in some sense still are an open issue, for the main focus has been on safety properties.

In this paper we have addressed the problem of (secure) interpartition communication in TSP architectures for aerospace systems. A set of communication scenarios was analysed, and solutions to the issues raised by them were proposed. The solutions range from (classical) software-based mechanisms, to innovative hardware-based mechanisms.

In future work we will further address these issues in order to ensure that secure communication is enforced, and that no further vulnerabilities are present.

References

[1] Airlines Electronic Engineering Committee (AEEC): Avionics Application Software Standard Interface - ARINC 653 Part 1 (Mar 2006)

[2] Almeida, K., Pinto, R.C., Rufino, J.: Fault Detection in Time- and Space-Partitioned Systems. In: 5th Simpósio de Informática (INFORUM). Évora, Portugal (Sep 2013)

[3] Alves-Foss, J., Oman, P.W., Taylor, C., Harrison, W.S.: The MILS architecture for high-assurance embedded systems. International journal of embedded systems 2(3), 239–247 (2006)

[4] Craveiro, J.P.: Integration of generic operating systems in partitioned architectures. Master's thesis, Faculty of Sciences, Univ. of Lisbon (Sep 2009)

[5] Daemen, J., Rijmen, V.: The design of Rijndael: AES-the advanced encryption standard. Springer (2002)

[6] Dworkin, M.J.: Sp 800-38b. recommendation for block cipher modes of operation: the CMAC mode for authentication (2005)

[7] OAR: RTEMS C User's Guide. Tech. rep. (2011)

[8] P. Fortescue, J. Stark, G.S.: Spacecraft Systems Engineering. Wiley (2003)

[9] Rufino, J., Craveiro, J.P., Schoofs, T., Tatibana, C., Windsor, J.: AIR technology: a step towards ARINC 653 in space. In: Proc. of the Data Systems in Aerospace Conf. (DASIA 2009). Istanbul, Turkey (May 2009)

[10] Rufino, J., Craveiro, J., Verissimo, P.: Architecting robustness and timeliness in a new generation of aerospace systems. In: Casimiro, A., de Lemos, R., Gacek, C. (eds.) Architecting Dependable Systems VII, pp. 146–170. Lecture Notes in Computer Science, Springer Berlin Heidelberg (2010)

[11] Rushby, J., Randell, B.: A distributed secure system. Computer 16(7), 55–67 (1983)

[12] Rushby, J.: Partitioning in avionics architectures: Requirements, mechanisms, and assurance. NASA Report CR-1999-209347 (Jun 1999)

[13] Watkins, C.B., Walter, R.: Transitioning from federated avionics architectures to integrated modular avionics. In: Digital Avionics Systems Conf., 2007. DASC'07. IEEE/AIAA 26th. pp. 2–A. IEEE (2007)

[14] Windsor, J., Eckstein, K., Mendham, P., Pareaud, T.: Time and space partitioning security components for spacecraft flight software. In: Digital Avionics Systems Conf. (DASC), 2011 IEEE/AIAA 30th. IEEE (2011)

[15] Windsor, J., Hjortnaes, K.: Time and space partitioning in spacecraft avionics. In: Space Mission Challenges for Information Technology, 2009. SMC-IT 2009. 3rd IEEE International Conf. on. pp. 13–20. IEEE (2009)