Preparing future Mission Data Systems for Secure Space Communications

Michael Koller[1], Max Pignède[1], Daniel Fischer[2] and Pier Bargellini[3]
European Space Agency (ESA) / European Space Operations Centre (ESOC)
Robert-Bosch-Straße 5
64293 Darmstadt, Germany

and

Àlvaro Manchado[4]
GMV
Robert Bosch Straße 7
64293 Darmstadt, Germany

The ESA Sentinels spacecraft constitute the first satellites responding to the Earth Observation needs of the GMES programme. In the case of the Sentinels, the need for Telecommand authentication and integrity has been a main design driver. All spacecraft are equipped with on-board units which ensure that no commands from any unauthorised source are accepted by the spacecraft.

The introduction of security features into the communications link has clear impacts on all elements involved in the communication. The basic design approach of ESA for new missions is to apply standards as far as possible. It is obvious that certain considerations need to be taken in order to allow for secure communications while still in line with the underlying standard.

Over the years, ESA has followed a strict re-use concept on the side of the ground data systems. The re-used generic components do not yet support secure communications and therefore need to be upgraded in this respect. In order to avoid problems with backward compatibility of software, special care has to be taken to ensure that the existing functionality implemented by the kernel remains mostly unchanged.

This paper presents all aspects of secure telecommanding introduced by the Sentinel spacecraft. The impact on the usage of the standard is addressed first, followed by the key aspects of the secure commanding. The operational concept of the Sentinel spacecraft is presented. Finally, the paper presents the challenges faced on the side of the mission data systems and how they are addressed.

I. Introduction

OVER the last years, the number of threats to a space mission's infrastructure has increased. This is mostly due to the use of open standards and the ease of access to space communication hardware. In parallel, space missions are becoming essential assets to the stakeholders once operational. As stakeholders rely on the availability of the mission products, the mission assets need to be protected from unauthorised access. Nevertheless, only little consideration has gone into implementing security measures for space missions until today.

Regardless of the potential security implications resulting from the high content of information in modern-day Earth Observation imagery and the thusly increased sensitivity of such data, the threat of active attacks on space missions is a threat which should and will be taken more seriously in the coming years. It is not overtly complex to manufacture and deploy a small communications antenna with sufficient transmission capability to connect to an orbiting satellite. Once this is achieved, the mission can be seriously harmed and jeopardised to the extent of a complete mission loss.

[1] Data Systems Manager, Earth Observation Mission Data Systems Section, ESA/ESOC.
[2] Data Systems Manager, Application and Special Projects Mission Data Systems Section, ESA/ESOC.
[3] (previously) Sentinels FOS Responsible, Earth Observation Mission Division, ESA/ESOC.
[4] Sentinels Mission Control System Development Project Manager, GMV.

While communication security is a well-established topic in terrestrial communication, not many implementations exist today for usage in space missions. Due to the specific nature of space missions, established technologies cannot be mapped directly on space communications. However, the basic principles are still valid and can thus be reused to secure communications.

One of the first ESA missions to actively consider those potential threats, also due to the importance of the mission products to the stakeholders, are the Sentinel spacecraft which are currently under preparation in order to serve the GMES needs for data acquired from space assets. As the ground infrastructure for the Sentinels is based on reusing existing software components, strong considerations have gone into the question how the security aspects can be added on top of the existing systems without jeopardising the possibility to continue incorporating future evolutions of the reused software.

The remainder of this paper is structured as follows. First, the Sentinels missions are briefly described, followed by the security measures which are implemented. Following this, the ESA approach of reusing existing software is discussed. The resulting changes to the existing software components and how those are handled are presented afterwards.

II. The GMES Sentinels

The main objective of the Global Monitoring for Environment and Security programme (GMES) is to support Europe's goals regarding sustainable development and global governance of the environment by providing high quality data, information, services and knowledge in a timely manner. To achieve this objective, the GMES programme features several components: In-Situ Measurement component, Data Infrastructure component, Service Provision component and the Space Segment component. The latter is responsible for the operational provision of Earth Observation data, achieved by a series of Sentinels spacecraft. The ESA Sentinels constitute the first series of operational satellites responding to the Earth Observation needs of the EU-ESA GMES programme. At the moment, ESA is tasked to launch the first set of Sentinels – Sentinel-1, -2 and -3 – and in the case of the first two is also tasked to operate the spacecraft. Each of the missions will finally be comprised of two spacecraft in phased orbits to guarantee the best possible coverage.

Sentinel-1 is a C-band interferometric radar mission for the continuation of SAR operational applications. Those two C-band radar satellites will provide continuous all-weather day/night imagery for user services, especially those identified in ESA's GMES service elements programme and on projects funded by the European Union (EU) Framework Programmes. Three priorities ("fast-track services") have been identified: marine core services, land monitoring services and emergency services. Compared to the current satellites in orbit, substantial improvements of data provision in terms of revisit frequency, coverage, timeliness and reliability of service are required. Services encompassing ship and oil-spill detection, wind speed measurements and sea-ice monitoring require daily revisits (mostly at northern latitudes) and delivery of data within an hour of acquisition. In contrast, land services involving interferometry and cover classification require global coverage every 2 weeks at most and consistent datasets.

Sentinel-2 is a multispectral high-resolution imaging mission for GMES land monitoring. This mission is tailored towards the needs of operational land monitoring and emergency services. Sentinel-2 will provide enhanced continuity of multispectral imagery provided by the French SPOT (Satellite Pour l'Observation de la Terre) series of satellites (for resolution of typically 10 metres and above). In order to meet the user requirements, Sentinel-2 satellites will provide imagery for the generation of high-level operational products such as land-cover maps, land-change detection maps, and geophysical variables, for example, leaf area index, leaf chlorophyll content and leaf water content.

Sentinel-3 is a mission with a wide-swath, low-medium resolution optical and infrared radiometers and a radar altimeter package. Sentinel-3 is an Earth observation mission carrying large swath/medium spatial resolution optical instruments and a radar altimeter. Sentinel-3 carries a set of 4 main payloads: an Ocean and Land Colour Instrument (OLCI), a Sea and Land Surface Temperature Instrument (SLSTR), a Radar Altimeter (RA), and a Microwave Radiometer (MWR). These are complemented by a GNSS receiver, a DORIS terminal, and a Laser Retroreflector.

As the mission products from the Sentinels are very important for all stakeholders, considerable effort has been put into analysing mission threats and into defining a concept to protect the space assets from potential attacks from ground. Those are presented in the next section.

III. Sentinel Communication Security considerations

In order to properly assess the security requirements for the Sentinels without having a negative impact on mission operations, a working group has been established including security experts and spacecraft operators. A detailed security analysis has been conducted, which is not repeated in great details in this paper. All general

threats on communications systems – such as eavesdropping, traffic analysis, spoofing and man in the middle attacks – have been analysed as well as threats which are specific to space missions – all as before plus the related implications. In addition to the potential security threats, the working group has also considered the impacts which any security measure may have on a space mission. In the end, it had to be ensured that the security implementation would not lock out the legitimate users. The major concerns have been found in the areas of performance and complexity on top of non-availability of data.

A. Security impacts on space missions

The operations required for a security scheme might be in conflict with the overall timeliness requirements or processing power of the system. Especially when considering real-time services, encryption and other security related functions might delay the delivery of data, rendering it useless in the end. There are extensive analysis on the performance of some AES hardware and software. It is shown that mere software implementation is not able to cope with the downlink data rates of a modern remote sensing mission. That means, that while the data can be prepared outside visibility and downlinked in a secure way, real-time operations are not supported. On top of this, current off the shelf spacecraft buses do not include any hardware module to perform security functions. As highlighted, this impact affects the large amounts of telemetry much more than the – normally modest amounts of – telecommanding but still has to be taken into account for overall performance considerations.

Complexity increases with every security measure implemented. One of the aspects of a security scheme which introduces the most complexity is the aspect of key management and the compatibility of the security implementation with existing systems. There are recommendation for some steps for successful key management for a space mission. Key management requires measures for secure generation, distribution and storage of keys throughout the lifetime of a security implementation. This usually requires a special key management infrastructure which needs to be compatible and interoperable with any other system required for the mission operation. All entities included in the communication have to be upgraded to support the new concept. When reusing existing infrastructure, the complexity is increased even further. This is especially true in the case of spacecraft hardware. Not many spacecraft already have provisions for any communication security, so any security unit must be integrated into the communication chain on-board the spacecraft without interfering with existing modules. An approach which is often used is to leave existing communication and data handling units unchanged and instead to patch the security modules into the communication chain. Such attempts often result in overly complex systems which require thorough testing in order to ensure proper operations.

B. Sentinels Security Concept

Combining the possible security threats and the security concerns, the working group devised a plausible secure operations concept which satisfies both, the need for security and the need for service continuity. It has been concluded, that the need for telecommand authentication and integrity is the most important for the missions. Those considerations have been an issue during the definition of the requirements of the GMES Sentinel missions at space and ground segment level.

The security concept aims at achieving telecommand authentication including anti replay protection and confidentiality. It comprises the authentication of telecommand segments sent to the spacecraft and ensures the confidentiality of certain cryptographic key information sent to the spacecraft. All cryptographic activities are carried out using symmetric encryption techniques. This decision has been made due to performance considerations.

Authentication shall ensure that the spacecraft only accepts commands which have been sent by a trusted entity, i.e. the control centres responsible for the spacecraft. This data origin authentication ensures data integrity at the same time; the information sent by a trusted entity has not been modified before reaching the spacecraft. Anti-replay protection has been put in place to guarantee that information sent by a trusted entity will not be accepted by the spacecraft if resent at a later point in time by a non-trusted entity. The second objective, confidentiality, shall ensure that no confidential information is disclosed to any third party which is not authorized to possess this information. In case of the GMES Sentinels spacecraft, confidentiality is only applied to sensitive information concerning the secure commanding concept (e.g. cryptographic keys that are uploaded to the spacecraft). All spacecraft are equipped with redundant on-board Authentication Units (AUs) which handle all security related functionality on board.

Authentication and anti-replay protection of the commands sent to the spacecraft is achieved by appending a Logical Authentication Counter (LAC) and a Message Authentication Code (MAC) to the Telecommand body. The LAC is basically an increasing counter. The MAC is generated by encrypting the hash digest of the command body and the LAC with a secret cryptographic key. LAC and MAC are referred to as security trailer.

Once the AUs on the spacecraft receive an authenticated command segment, they repeat the same procedure with the active key, including the uplinked LAC in the hash calculation. If the resulting value matches the MAC appended to the segment, the command segments is considered to be sent by a legitimate source and has not

been tampered with. Otherwise, the command segment is not accepted by the AU and commands inside are not executed.

One major question which the security working group had to answer once the overall concept was defined was the question, where in the communications stack authentication should take place. It has been decided to locate the security aspects between the data-link and the segmentation layer on the CCSDS protocol stack. This ensures that the SLE services remain untouched, a fact which makes the security operations transparent to the ground stations and other relays. This means, that all commands targeted to the Sentinel spacecraft will be embedded in Command Segments which have a LAC and MAC attached. This is depicted in figure 1.

Figure 1 also highlights the fact that the identifier of the Multiplexer Access Point (MAP ID) on-board the spacecraft can drive the selection of the correct cryptographic key for the MAC generation. This is important as the on-board authentication units require special keys with respect to the other on-board systems.

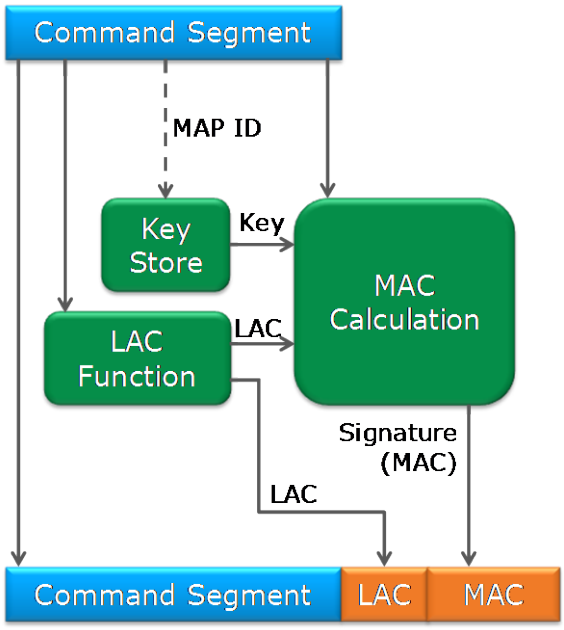

Figure 1. Authentication of a Command Segment.

The only information to which confidentiality is applied is cryptographic material itself. This will be discussed in the next section.

C. Key management considerations

As for all secure communication protocols – regardless if on ground or in space – one of the major challenges is key management. Key management is a mandatory security support functionality. It comprises all activities concerning the generation, distribution and synchronisation of cryptographic keys. Without proper key management in place, no secure communications protocol can perform correctly. The most challenging aspect of key management is most often the fact that confidential information has to be transported via non-secure channels. In case of a space missions, it is simply impossible to physically reach the spacecraft to distribute new keys. On the other hand, it is not feasible to distribute all required keys before the mission start as the number of keys required is very high. Therefore, the secure operations concept needs to address the point of replenishing communication keys via a normal SLE space link channel.

Generally, there are two key-management mechanisms known: static and dynamic key-management. Static key management refers to the principle, that all cryptographic keys which are required for the mission are distributed to the involved parties before the mission commences, i.e. before the launch. As indicated above, this approach is not very feasible. First there is the number of keys to be considered. Assuming the mission lifetime of seven years plus allowing for potential extensions, the number of keys becomes fairly large, depending on the key change interval. A large number of keys however requires more memory on-board. Secondly, the number of keys could become the limiting factor for the mission lifetime, or the fact that keys are running out might lead to shortened key changing intervals which might impair mission security.

Dynamic key-management means that keys are exchanged/distributed when the mission is already operational, i.e. in flight. As stated above, this approach is very problematic as it requires the keys to be exchanged over the space link channel, a communications channel which is not secure as it is in a normal mission. Any attacker could eavesdrop on the key exchange and use the key to send bogus commands which would be accepted on-board.

Apparently, neither of the two approaches is feasible for a space mission; luckily, there exists the possibility to use a hybrid approach, which is what has been decided for the Sentinels. In order to implement the hybrid approach, a two-level key hierarchy has been devised: there are so-called master keys which are handled as static keys (i.e. they are put on-board the spacecraft prior to launch) and there are session keys which are dynamic keys (i.e. they are uplinked to the spacecraft once it is in flight). All keys are considered sensitive information; therefore their confidentiality has to be preserved. The purpose of the master keys is to establish a secure channel over which the dynamic session keys can be exchanged, therefore ensuring the confidentiality and integrity of the session keys. They are generated before launch and stored in a secure, read-only area of the on-board memory such that they cannot be altered once the spacecraft is flying. The session keys are used for authenticating the commands sent to the spacecraft as indicated in figure 1. They are generated and uploaded regularly during the mission lifetime.

Figure 2 depicts how new session keys are uplinked without compromising their integrity. Session keys which are supposed to be uploaded are encrypted using a master key. The cipher is then used as command parameter of a normal command which is sent to the spacecraft and is authenticated normally.

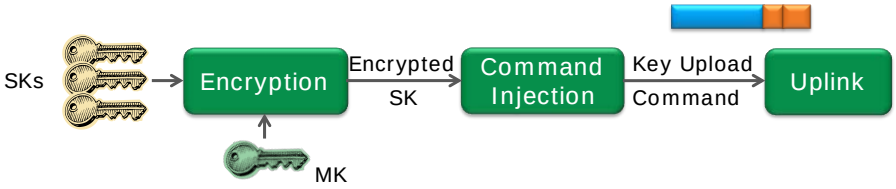

Figure 2. Uplinking Session keys (SK) with the use of Master keys (MK).

In addition to the usage of master keys for encrypting keys for the upload, the master keys also play a crucial role in contingency recovery. There is always a master key active to be used for authentication recovery, meaning that if for some reason one of the active session keys gets corrupted, nominal service can be re-established by the use of this authentication recovery master key. This is one of the measures put in place in order to avoid that the security implementation might negatively impact the mission.

Additionally, the spacecraft operators can at any time enable or disable the secure commanding functionality; if it is disabled, all commands are uplinked without authentication. Commands addressing the authentication units are only accepted on-board when authenticated regardless of the state of the secure commanding functionality. This ensures that nobody can negatively influence the authentication units while the spacecraft is operated without the secure commanding functionality.

This concludes the brief look into the Sentinels secure operations concept and the short explanation of the design. While at this point in time, quite many parameters are well defined and implemented (e.g. the number of master keys each authentication unit can store, size of the keys, …), some operational decision have yet to be made, e.g. the key replacement interval for session keys (once per command vs. once per commanding pass or once per day/week) and for master keys (one dedicated master key for each set of sessions keys that are encrypted vs. one master key for each full uplinking campaign). A well-balanced trade-off has to be found between security, resources (i.e. time for commanding passes, available master keys over the mission lifetime, …) and operability. The positive aspect is that the current hybrid approach of static and dynamic key management is allowing for a great degree of flexibility.

Once the operational concept was understood, the impact on the mission implementation had to be assessed and addressed. On the side of the ground software, this was especially challenging, as ESOC's ground software is not developed from scratch, but reuses heavily existing ground software components. This is highlighted in the next section.

IV. Reusing existing software

Very often, space missions have a huge set of commonalities. Those could be the same sort of orbit profile, the same set of communication constraints or the same characteristics of the spacecraft bus. Naturally, spacecraft designers exploit those commonalities by applying knowledge acquired in earlier missions to current missions in preparation. Obviously this helps to cut development time and costs. Not only does this lead to cost savings but there is also room to apply lessons learnt, thus improving the quality of the spacecraft and the mission. ESOC came to the conclusion that similar synergies can be achieved on the side of the Mission Data Systems, which are the Mission Control System and the Operational Simulators.

Most of ESOC's activities involve science and Earth observation missions. Such satellites are complex and every mission is different. At first sight, this seems to prevent the reuse of tools and functions between missions, but software reuse has been achieved using various techniques, such as changing configuration via setting of different parameters in tables or databases and using appropriate software engineering technologies.

Thanks to the extensive experience and the long period of applying the software reuse strategy, ESOC was able to collect a large amount of common functionalities among missions that are available today in form of infrastructure software. They are bundled into the ESA Ground Operational Software package EGOS which offers infrastructure components for virtually almost all parts of a Spacecraft Ground System. Two parts of EGOS are particularly relevant in the realm of Mission Data Systems:

- SCOS-2000 is the basis for mission specific mission control systems and covers most of the functions required for telemetry reception and processing, telecommand uplink and verification, data archiving, display and retrieval, and data distribution;
- the Simulus toolset is the basis for mission-specific Operational Simulators. It covers the SIMSAT simulator kernel and the Generic Models, including on-board processor emulators, orbit prediction

and propagation (including environment perturbations), selected on-board subsystem models (such as data handling, thermal, electrical) and ground models.

Every mission launched or operated by ESOC is attempting to reuse as much of those shared infrastructure elements, and over the years, considerable cost savings have been achieved. Despite all the commonalities between space missions, there can still be a great amount of differences if considering different mission families, e.g. earth observation missions vs. planetary missions or navigation missions. Whilst all members of one family share an even greater degree of commonalities, differences to other missions limit the possibility to reuse existing systems. In order to address this limitation, ESOC as introduced various mission family kernels on the side of the Mission Control System, which further capture the commonalities inside the mission. One typical example for a earth observation speciality which is not shared with other missions is the delayed execution of commands loaded on-board depending on the orbital position of the spacecraft, not based on time. The inclusion of the so-called Earth Observation Mission Control System kernel has led to even greater degrees of software reuse.

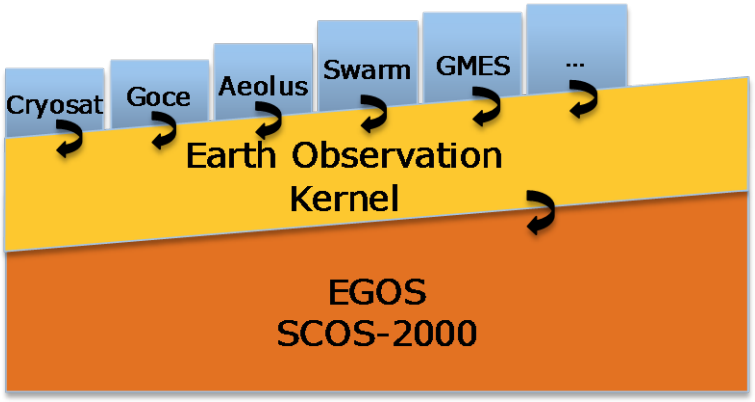

Figure 3. ESOC reuse of common software elements.

Figure 3 depicts the basic idea behind this reuse. There are several aspects to be taken into consideration.

For the full development cycle, ESOC is following a delta approach. This means, that every layer of software is defined only as a delta with respect to underlying elements. For example the Software Requirements Specification for a mission control system for a given mission does not repeat any of the functionality supplied by the reused elements, but only states the differences. This philosophy is followed all the way, through design, implementation and testing and verification.

To give one example for the impact of the delta approach to number of requirements: 82% of the Aeolus mission requirements for the Mission Control System are covered by the generic SCOS-2000 kernel, 15% are covered by the Earth Explorer Kernel and only 3% are mission specific. On the simulator side there clearly are some more mission specific components as some on-board modules (mostly the payload but also some modules of the bus) might never have been simulated before. Unfortunately here it is not as easy to adapt generic software through configuration changes. The Aeolus simulator requirements are covered for 63% by the SIMSAT kernel, 18% are covered by Generic Models, 11% are covered by requirements common to the Earth Explorer Family and the remaining 18% are mission specific.

The savings in effort are carried forward through the software lifecycle. Looking at the design documents as presented in figure 4, the effect becomes immediately obvious. The advantages are clear: the mission experts can focus on the new aspects of the missions during reviews and testing and do not need to invest great efforts into re-examining what is known already.

System	Size of ADD
SCOS-2000 ADD	approx. 3000 pages
Earth Explorer ADD	approx. 1000 pages
Aeolus MCS ADD	approx. 120 pages

Figure 4. Pages in Architectural Design Document (ADD).

The reused software components are living systems. They evolve over the time in two aspects: maintenance and incorporation of new requirements. When the EGOS components are maintained by a central point – ESOC infrastructure – synergies can be achieved as not every mission needs to pay for the fixing of the same software issues. Additionally the testing effort is much more widespread amongst the missions. Furthermore, different missions might test different areas of the common components in great depth, resulting in an overall improved and more stable system. The same is true for mission family kernels.

Apart from the day-to-day maintenance, the kernels need to evolve in order to satisfy future mission needs. In regular intervals, mission (family) specific implementations are evaluated if they can be applied to future missions. If new common requirements are identified, those can be incorporated into the common components. This is indicated by the downward facing arrows in figure 3 and the illustration of the growing common components.

This concludes the short excurse into the ESOC practice of reusing and sharing basic building blocks for mission data systems. Over many missions, ESA has successfully applied this delta scheme in the realm of mission data systems and has therefore achieved impressive results in terms of software quality, cost of the system and short development schedules. Having said all this, and coming back to the main topic – secure operations for space missions – the final question is how the usual practice of software reuse is going together with the inclusion of new requirements concerning the secure commanding. This is discussed in the following chapter.

V. Upgrading software to cope with secure operations

As described in previous sections of this paper, the GMES Sentinels are going to implement command authentication while at the same time following the standard ESOC approach of reusing common infrastructure elements for mission data systems. The mission data systems under consideration in this paper are the Mission Control System (MCS) and the operational spacecraft simulator (Sim).

The MCS is a software system that enables the operators on the ground to interact with the satellite. It receives, interprets, analyses and archives telemetry, the data downlinked from the spacecraft used for monitoring its health and receiving its mission products. The MCS also generates, verifies and uplinks commands, transmitting instructions to the satellite to control all its operations. In addition to these functions, the MCS usually provides support for mission planning, data analysis and distribution, telemetry and command database management and on-board software management.

The Operational Simulator is a software system that simulates the satellite, the ground stations and, to a certain extent, the space environment. An Operational Simulator is developed for almost every mission operated by ESOC. It is used to support testing of the ground systems, including the MCS and to prepare and support all the validation tests and simulation campaigns which serve to test and validate the flight operation procedures as well as to train the ground team for routine operations and possible contingencies.

Keeping in mind that the reused system kernels are providing most of the common functionality, it becomes obvious that changes to the commanding link – i.e. the inclusion of the LAC and the MAC – and to the data-handling system are necessary. This section describes the changes implemented for the Sentinels with two goals in mind: satisfying mission requirements and keeping changes to the underlying software minimal in order to not impede further inclusion of future releases and patches.

D. Mission Control System

As described above, the Mission Control System (MCS) is for a great part reusing existing implementations from the SCOS-2000 kernel and the Earth Observation Mission Control System kernel. As the MCS is a distributed system with a server/client architecture which is able to support numerous clients, there are multiple tasks involved in the dispatching, encoding and releasing of a command to the spacecraft. Those are illustrated in figure 5 in the boxes on the left hand side.

The command sources on the clients are the interface where the operators are constructing the command which is to be sent to the spacecraft. When the command is dispatched from the command source, it is sent to the central MCS server which first of all does the multiplexing from the various sources and passes the command packets on to the Command Releaser. There, the packets are put into segments and are encapsulated in CCSDS Command Link Transfer Units (CLTUs) which in turn are relayed to the ground station via a network interface system (not depicted) for uplink to the spacecraft.

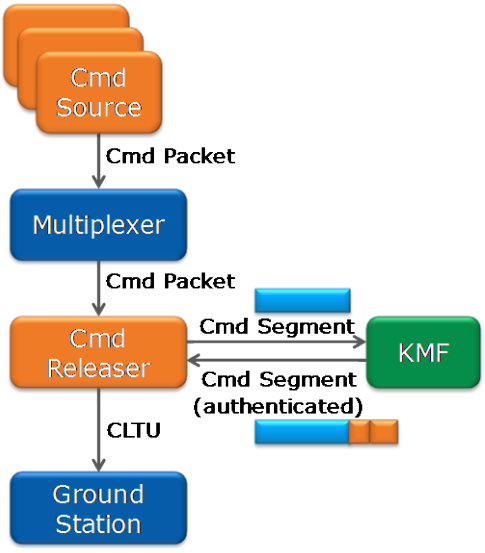

Figure 5. Elements involved in MCS Command release.

The challenge to integrate the security functionality would be easily addressed, if any of the existing server tasks were exchanging command segments. It would have been rather simple to plug an additional process into the chain which can perform the activities required for command authentication. Unfortunately this is not the case.

First of all, a new task – the Key Management Facility (KMF) – has been introduced to handle all activities concerning the real-time command authentication. This plugging in of a new task can be done with no disturbance to the underlying structures. The KMF receives a readily built commanding segment, calculates LAC and MAC based on some information (as depicted in figure 1) and sends the command segment with the appended security trailer back to the command releaser for further processing.

Additionally, some changes have to be applied to the existing applications: the command sources needed some minor tweaking in order to support secure commanding, e.g. an indicator of the current operational mode, etc. The bigger changes have been applied to the Command Releaser task. The releaser has to make a decision for each command segment, if it has to be routed to the KMF for inclusion of a security trailer. This decision tree is illustrated in figure 6. Basically, the command releaser knows two basic operational modes from a security point of view: security on or security off.

This switch has been added for reasons of making the system future proof. As indicated earlier in this paper, common functionality which has been developed for a mission can be repatriated into underlying kernels in order to make this functionality available to future missions. The secure commanding functionality is such a candidate functionality. However, not all future missions might implement command authentication, therefore they can very easily disable all of this functionality without negative effect to the mission.

The releaser has first to check the general set up (overall support of secure commanding or not). If no, the path has ended and the segment is sent out for release. Otherwise, it has to decide if currently the system is set up for secure commanding (this means that the security trailer is appended to all commands) or not (in this case, only commands addressed to the authentication unit have to be authenticated as indicated earlier). If in secure, the segment passes to the KMF and is authenticated. If the system is currently in clear, the Releaser has to check the recipient of the command segment, based on the identifier Multiplexer Access Point (MAP ID) in the segment. It the MAP ID designates the authentication unit, the segment must be routed to the KMF even if in clear, as all commands addressed to the authentication unit have to be authenticated. If the segment does not need to have any security trailer appended, the releaser finally needs to check if a default authentication trail is configured to be appended (as it might be expected by the spacecraft in any case) and act accordingly. Finally, the command can be released to the Network Interface System (NIS).

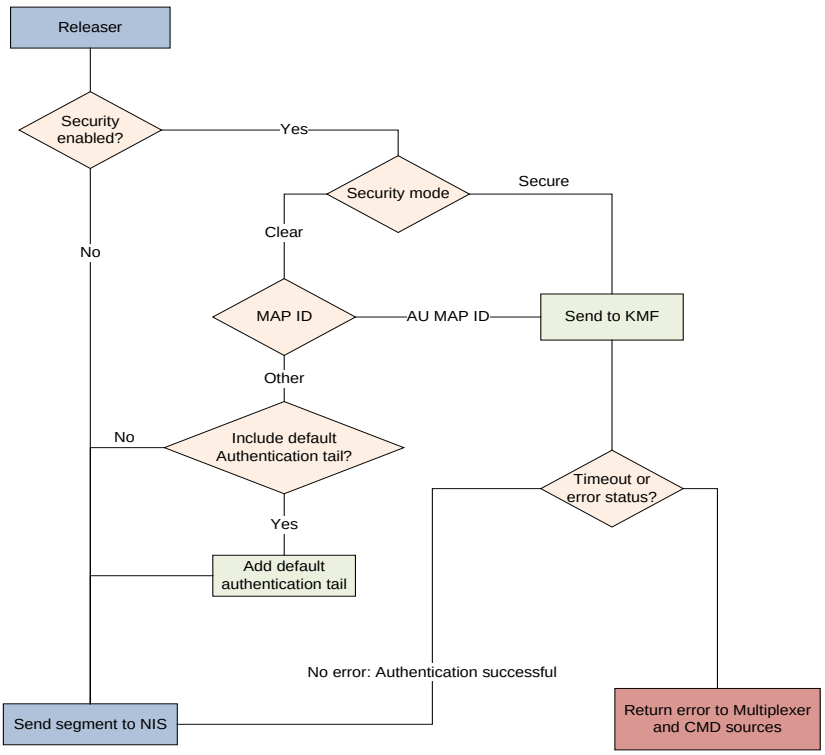

Figure 6. The path for a command segment.

Overall, the changes required in the existing command releaser are all illustrated in figure 6. No further changes are necessary, because – as indicated before – the actual security functionality is contained in the KMF. So the changes to the MCS software is minimum and quite contained from a source code point of view. This is because it will be rather simple in the future to include new versions of the reused software.

In addition to the changes on the side of the MCS, a new facility has been developed – the Master Key Management Facility (MKMF) – which is in charge of all aspects of key management (storing, archiving, securing) and key generation. This facility is completely decoupled from the MCS, only encrypted files are passed to the MCS in an offline manner. Therefore, the MKMF is not part of the discussions in this paper.

E. Operational Simulator

The operational simulator is based on the SIMSAT infrastructure. The SIMSAT main features are the provision of a graphical user interface, the scheduling of simulation models, the logging of events and faults, the visualization of model data and the provision of user facilities to control the simulation. To ensure realistic behaviour, the simulator runs an image of the On-Board Software of the satellite and also models the

environment for position determination and attitude control. In addition the simulator includes specific functional models of all hardware equipment of the satellite.

SIMSAT is split into a number of Kernel components which load and execute a simulation and into a Man Machine Interface which allows monitoring and control of a running simulation. SIMSAT is the generic environment supporting the execution of a satellite simulator. The simulator is in charge of modelling the satellite behaviour. Its architecture is based on a reusable reference architecture and accommodate the addition of any specific models required in the simulation. The components and various elements are depicted in figure 7.

As indicated previously, the Sentinel spacecraft will implement a scheme for authenticated commanding, which includes a logical authentication counter (LAC) and a message authentication code (MAC) being appended to the commanding segment. This has implications for the simulator as well.

When a TC reaches the satellite, the on-board Authentication Unit (AU) regenerates the MAC for each authenticated segment and compares it to the MAC provided in the received commanding segment. The received command is only accepted if both MACs match. Then the authentication unit transfers correctly authenticated commanding segments further to the spacecraft systems.

Of course, the above secured commands reception process taking place on the satellite must be modelled in the simulator where in particular the AU model shall support the authentication of commands and shall include the same authentication algorithm as on the real spacecraft. In case of a failure, the effect in the simulator of the failure shall be identical to the effect of a failure on the real satellite. The simulator shall also be capable of responding to control commands addressed to the AU, including commands to load new Session Keys.

The software in charge of this modelling is located in the Data Handling subsystem (see figure 7b) and specific software will have to be produced based on the ESOC generic infrastructure. A further task required for the Sentinels is to extend the SIMSAT Generic Model SIMPACK (see figure 7b) in order to handle the optional

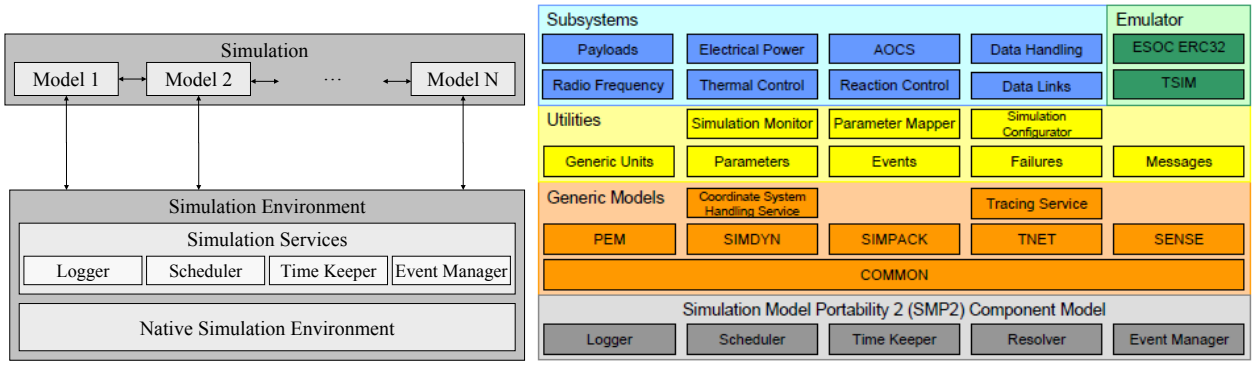

Figure 7a and b. Simulator components and elements.

PUS Authentication Tail (compliant to ECSS TC standard) carrying the Key Identification Field, the LAC and the MAC.

This concludes the excurse into the actual implementation. It becomes obvious that the design of the Sentinels mission data systems has been realised with a maximum of reusability in mind (e.g. the configuration aspects on the MCS side) and have been realised with only little impact to the underlying software. The actual application of this scenario is still outstanding, but all stakeholders are confident that the impact of including new software underneath the Sentinels implementation is feasible with little effort.

VI. Conclusions

As it has been presented in this paper, the GMES Sentinel spacecraft are the first ESA Earth Observation spacecraft to implement communications security on the commanding link. It has been decided to secure the spacecraft from unauthorised commanding access by adding a security trailer to the command segments which are sent to the spacecraft. The trailer is composed of a Logical Authentication Counter and a Message Authentication Code. The latter is obtained by performing cryptographic encryption of the hash value of the command segment and the Logical Authentication Counter. Only parties in possession of the right key can perform this operation in a way that the command segment is accepted by the spacecraft.

The challenge of key management and distribution is addressed by using a hybrid scheme of dynamic and static keys. The static keys are used as key encryption keys for uploading new dynamic keys and therefore serve as a means to establish a secure channel over which the dynamic keys can be shared between ground and the spacecraft.

Once the operational concept was clear, ESA addressed the challenge of implementing the new security features in its Mission Control Systems and Operational Simulators in a way which does not conflict with the paradigm of software reuse. The design of the Sentinels mission data systems ensures that the software components which are used to build up mission specific simulators and mission control systems have to be changed only to a very limited extent and that all changes are kept at a level which does not impede with the incorporation of newer infrastructure versions. Furthermore, configurability was a key design element when developing the new functionality, in order to make the implemented solution available to future missions as well with none or only little implementation effort.

At the current point in the project it can be concluded that so far the implementation looks well in line with mission requirements. First communication attempts have been successfully completed with the real spacecraft hardware. The aspects of reintegration of new infrastructure baselines or passing the system on to future missions has not yet been put to the test, but the design of the software does not raise any concerns in this area.