Requirements Definition for Onboard Data Systems for Life-Cycle Support and Management of End-to-End Security

Final Presentation

FP Days – ESTEC
Noordwijk, May 21-22, 2014

Airbus Defence and Space: Luc Planche, Jean-Paul Blanquart, Gilles Herrgott, Jean-François Soucaille, Dave Thomas
DSI: Lazslo Hinsenkamp, Marc von der Wall
KU Luven COSIC: Dave Singelée, Roel Peters

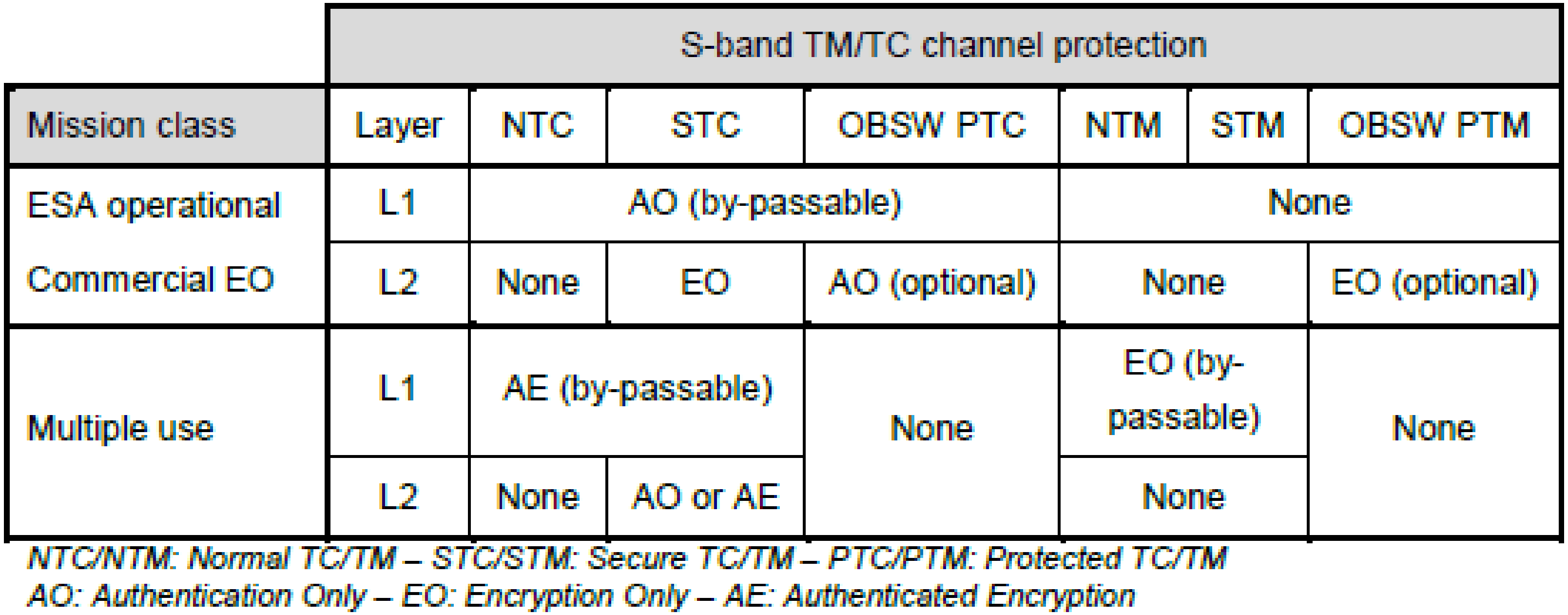
ESTEC: Marco Rovatti

Overview of the study

- Study objectives
  - propose a set of general recommendations for a security engineering process that aims at protecting the system development, and not the system itself, against aggressions.
  - detail recommendations for the management of security services: functional (keys management) and non-functional (FDIR)
- Main tasks
  - Review of Avionics Architectures and Security Approaches
  - Life-Cycle Security Engineering Process Analysis
  - Security Keys Handling/Loading
  - FDIR and Redundancy of Security Units

Review of current on-board security approaches

- Two major classes of protection, implemented at various protocol layers
- Various symmetric keys management schemes

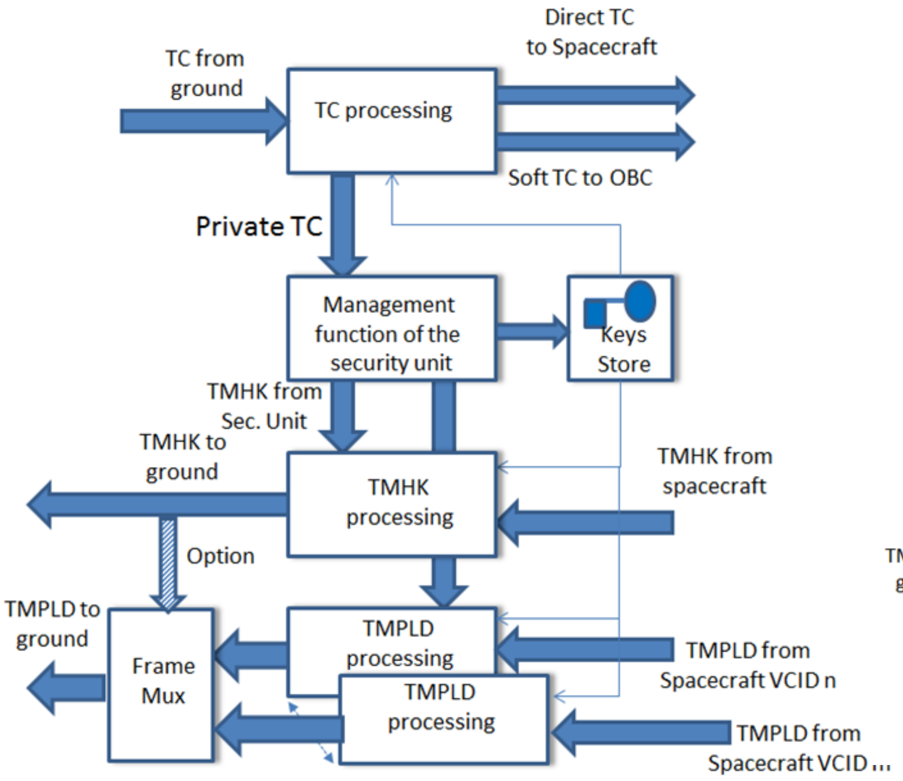


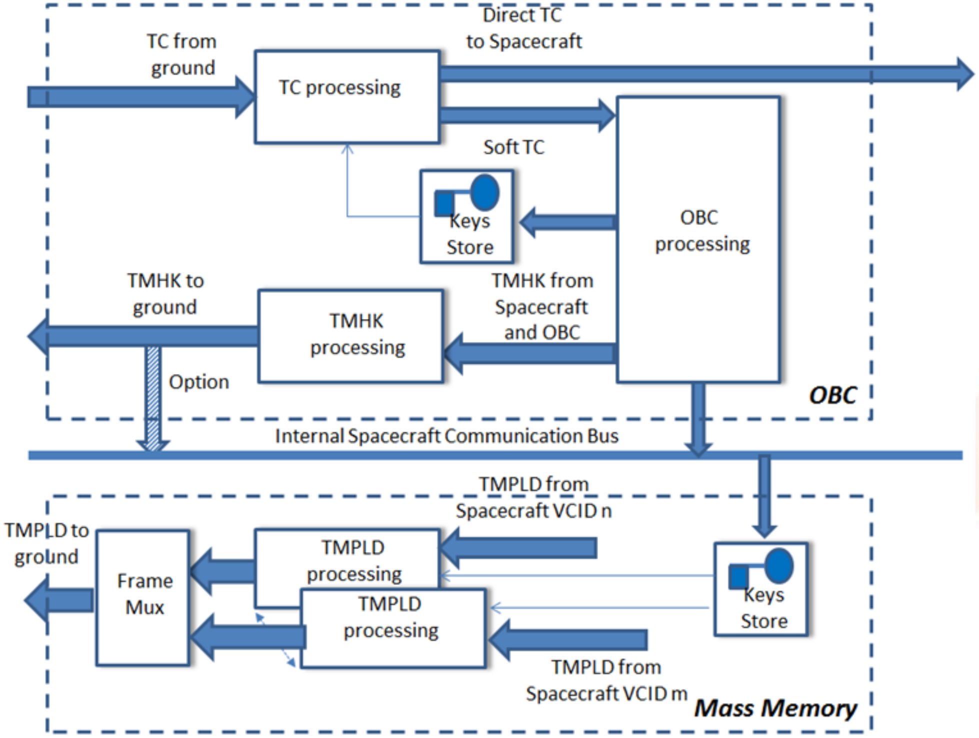
	S-band TM/TC channel protection						
Mission class	Layer	NTC	STC	OBSW PTC	NTM	STM	OBSW PTM
ESA operational Commercial EO	L1	AO (by-passable)			None		
	L2	None	EO	AO (optional)	None		EO (optional)
Multiple use	L1	AE (by-passable)		None	EO (by-passable)		None
	L2	None	AO or AE		None		

NTC/NTM: Normal TC/TM – STC/STM: Secure TC/TM – PTC/PTM: Protected TC/TM
AO: Authentication Only – EO: Encryption Only – AE: Authenticated Encryption

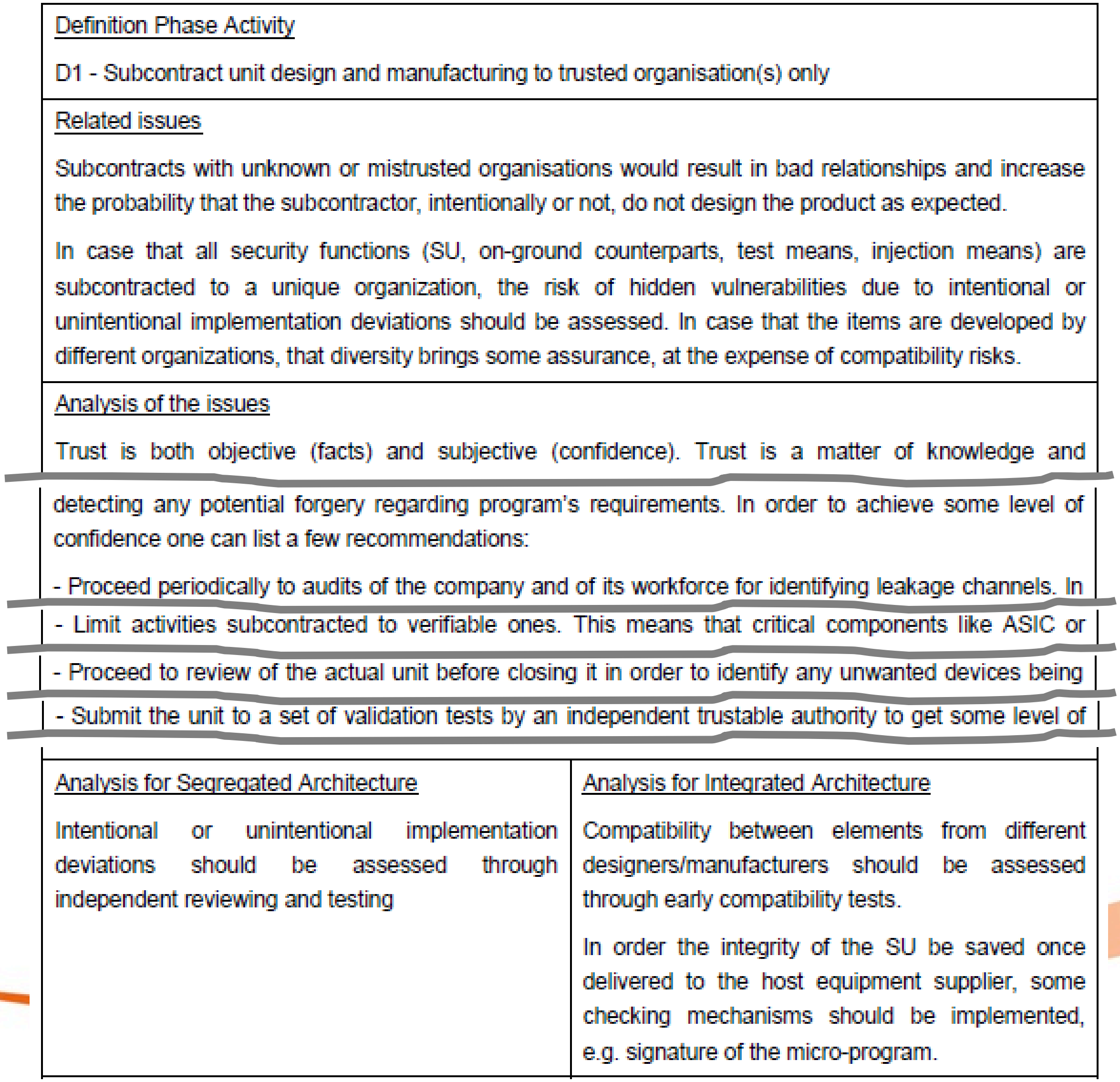
Reference implementations

- Segregated architecture vs Integrated architecture





Analysis of the life-cycle security engineering process



Definition Phase Activity

D1 - Subcontract unit design and manufacturing to trusted organisation(s) only

Related issues

Subcontracts with unknown or mistrusted organisations would result in bad relationships and increase the probability that the subcontractor, intentionally or not, do not design the product as expected.

In case that all security functions (SU, on-ground counterparts, test means, injection means) are subcontracted to a unique organization, the risk of hidden vulnerabilities due to intentional or unintentional implementation deviations should be assessed. In case that the items are developed by different organizations, that diversity brings some assurance, at the expense of compatibility risks.

Analysis of the issues

Trust is both objective (facts) and subjective (confidence). Trust is a matter of knowledge and

detecting any potential forgery regarding program's requirements. In order to achieve some level of confidence one can list a few recommendations:

- Proceed periodically to audits of the company and of its workforce for identifying leakage channels. In

- Limit activities subcontracted to verifiable ones. This means that critical components like ASIC or

- Proceed to review of the actual unit before closing it in order to identify any unwanted devices being

- Submit the unit to a set of validation tests by an independent trustable authority to get some level of

Analysis for Segregated Architecture	Analysis for Integrated Architecture
Intentional or unintentional implementation deviations should be assessed through independent reviewing and testing	Compatibility between elements from different designers/manufacturers should be assessed through early compatibility tests. In order the integrity of the SU be saved once delivered to the host equipment supplier, some checking mechanisms should be implemented, e.g. signature of the micro-program.

- An analysis of the issues relating to the security engineering process have been performed at system and security unit supplier level
- Additional activities and roles have been proposed wrt the traditional design and operation processes

In summary…

- The trustability of a unit is a chain, any not trustable entity acting in it will make it not trustable as well.

- It is meaningful to use a sub. company only for units dealing with security aspects under the responsibility of its own country in such a way that any further security question remains confined locally.

- The main question is how to insure that the trusted ‘component’ remains safe after being integrated by an external entity

Security engineering objectives & measures (1/2)

- Proceed periodically with audits of the company and of its workforce for identifying leakage channels.
- Limit activities subcontracted to verifiable ones. This means that critical components like ASIC or FPGA’s shall be provided by a trusted entity and shall be identifiable to avoid any substitution or unwanted modification.
- Proceed to review of the actual unit before closing it in order to identify any unwanted devices being added within the unit.
- Submit the unit to a set of validation tests by an independent trustable authority to get some level of confidence

Security engineering objectives & measures (2/2)

- Avoid the introduction of hidden channels in the control path of the unit.
- Inspect hardware for insuring direct link between trusted item and outside world.
- Provide a direct link between the trusted item and the outside world for allowing direct check of it. Could be conflicting with FDIR.
- If the unit is really able to endanger the system, use a second unit from an independent manufacturer in parallel.

Impact on ECSS

- Analysis of proposed security engineering process, activities, identified issues

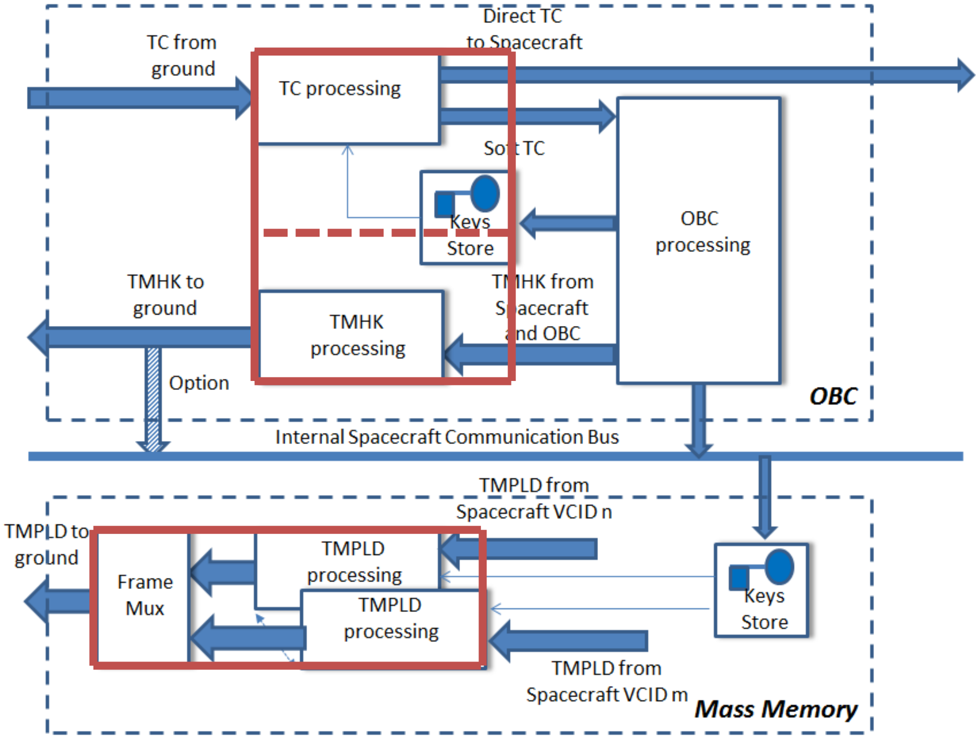
- Position with respect to relevant ECSS (Q30 (dependability), Q40 (safety), Q80 (software product assurance))
  - Possible conflicts
  - Possible lacks, recommendations, proposed complements

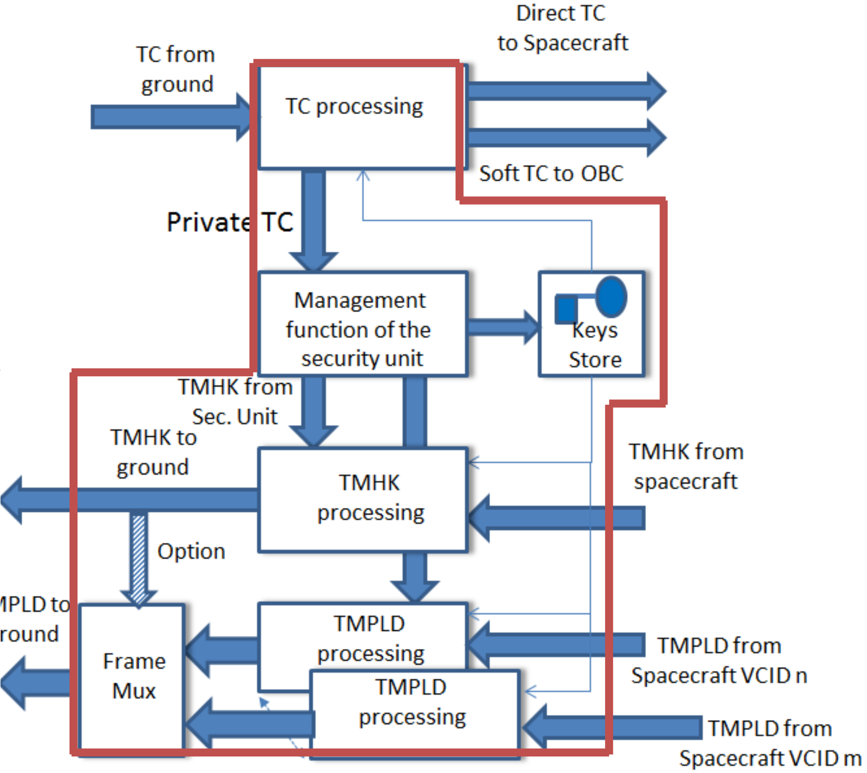
Impact on ECSS – main results

- No conflict identified
- However, security should be more explicitly addressed, and additional guidance provided on how to properly cover security within the general safety, dependability, product assurance and software processes
  - When and which security analyses should be performed
  - Coordination with dependability, safety analyses
  - Consolidation into global assessment
  - Highlight interest of existing means and techniques for security (e.g., product assurance rules against malicious code or other)

Impact on DHS - Approach

- Identification of security domains
- Analysis of interfaces between secure and un-secure areas





ACE

DSI Informationstechnik

KATHOLIEKE UNIVERSITEIT LEUVEN

COSIC

Impact on DHS - Synthesis

- Recommendations about Integrating and operating secure building blocks into a non-evaluated avionics
  - Interface characterisation
  - Assessment of available security units
  - Assessment of impacts on
    - budgets (mass, power, reliability, …)
    - Redundancy scheme, cross-strappings, dependability (safety) – security trade-off
    - Power supply
  - Security Unit management
  - Accomodation in the satellite
  - Qualification, validation (and links with satellite validation strategy)

Keys Management Solutions in ESA Space Missions

- Life cycle of spacecraft
  - □ Assembly, Integration and Test phase
  - □ Launch campaign phase
  - □ Early-orbit phase
  - □ **In-orbit phase**
  - □ End-of-Life phase

- Key management relevant in all phases
- Particularly important:
  - □ (Master) Key generation + secure transfer
  - □ Key seperation: key identifiers + cryptoperiod!!

General recommendations

- Protection of root of key hierarchy is crucial
  - □ Potentially onboard side-channels / Hardware Trojans
  - □ Physical protection of spacecraft before launch
- Cryptoperiod needs to be clearly defined
  - □ See Ecrypt/NIST/… recommendations
- Use dedicated keys to transport session keys
- Key separation is important
  - □ Key identifiers / tags
  - □ Prevents key swapping
- Authenticated encryption offers several benefits

Specific comments related to Sentinel 2

- No key confirmation
  - □ Ground station has no assurance that spacecraft possesses correct session keys
- Session keys for secure TCs are used to control master key activation -> use dedicated (master/recovery) keys instead

Two recommended key management solutions

- Symmetric-key-based solution
- Public-key-based solution

- Alternative (future) suggestions/ideas
  - □ Onboard session key generation
  - □ One-sided Diffie-Hellman key exchange

Recommended symmetric-key-based solution

- Key hierarchy
  - □ Master keys
  - □ Session keys normal TCs
  - □ Session keys secure TCs
  - □ Recovery keys to activate new master key
- Session keys are used explicitly in TM
  - □ 3 possible strategies
    - □ Compute MAC on each received session key
    - □ Confirme activation of new session key by MAC
    - □ Compute MAC only on first session key in received block
- Use 32-bit counter for all space missions

Recommended public-key-based solution

- Security against eavesdropping attacks
- Based on ECC Diffie-Hellman
- Key hierarchy
  - □ Authentication keys to authenticate DH messages
  - □ Session keys normal TCs
  - □ Session keys secure TCs
- Secure onboard random number generator needed in spacecraft
- Session keys still need to be confirmed by ground station

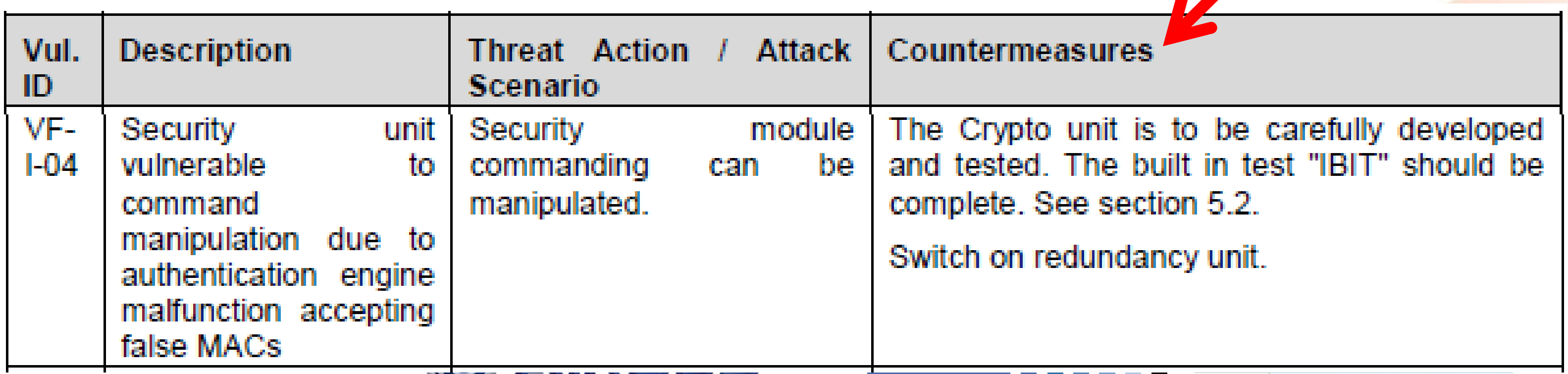
FDIR and Redundancy of Security Units

- Identify reference failure and attack scenarios
- Analyse redundancy schemes and FDIR, propose recommendations
- Split in two parts
  - Security Unit level
  - Functional level

Security unit level (bottom-up) analysis

- Generic Security unit made of usual components
  - □ control FPGA, crypto FPGA, oscillator, interfaces, reset circuitry, EEPROM ,RAM and power supply
- Identification of security/cryptographic consequences of the unit failure modes (new vulnerabilities)
- Elicitation of specific counter measures, in addition to "normal" failure handling

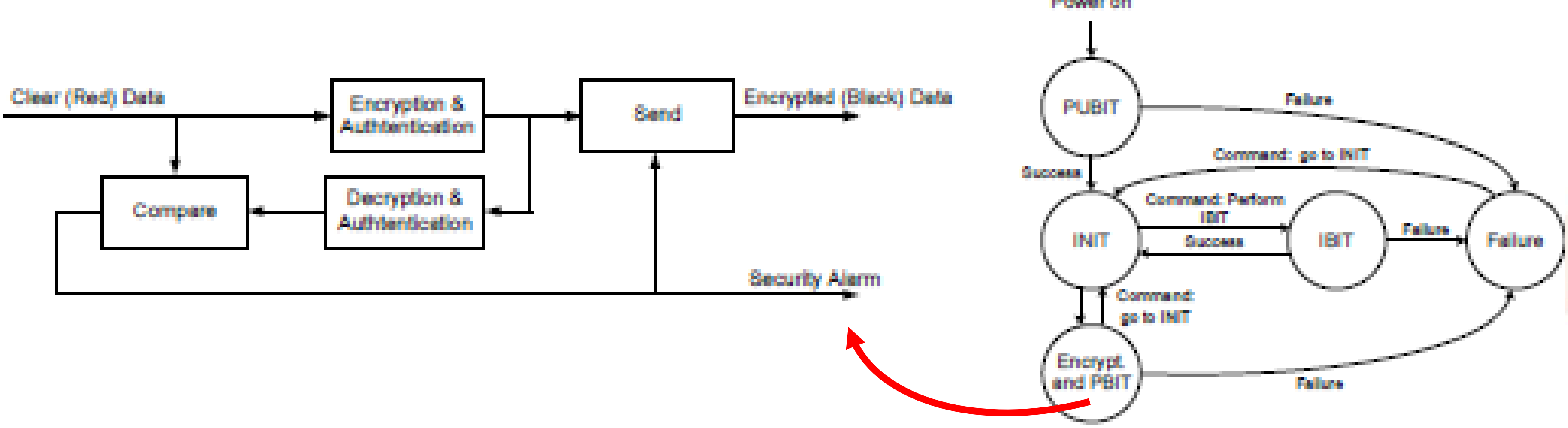
Block/ Component	Example Function	Failure Modes	Failure Cause	Example Local Failure Effects	Example System Failure Effects	Vul. ID
Control interface's authentication engine	Authentication of security unit commands	SC, OC, Stuck low/high	EEE Part Failure	TC accepted even if MAC check fails	Security unit vulnerable to command manipulation attacks	VF-I-04



Vul. ID	Description	Threat Action / Attack Scenario	Countermeasures
VF-I-04	Security unit vulnerable to command manipulation due to authentication engine malfunction accepting false MACs	Security module commanding can be manipulated.	The Crypto unit is to be carefully developed and tested. The built in test "IBIT" should be complete. See section 5.2. Switch on redundancy unit.

Impact on Security units

- Basic countermeasures recommended against vulnerabilities opened by design
  - □ Separation of red (unprotected data) and black (protected data) areas
  - □ No security by-pass.
- Mitigation measures and FDIR procedures shall be taken for dealing with the failures and vulnerabilities.
  - □ A general approach is to implement a set of built-in-tests



- The cryptographic keys management shall be made consistent with the redundancy scheme

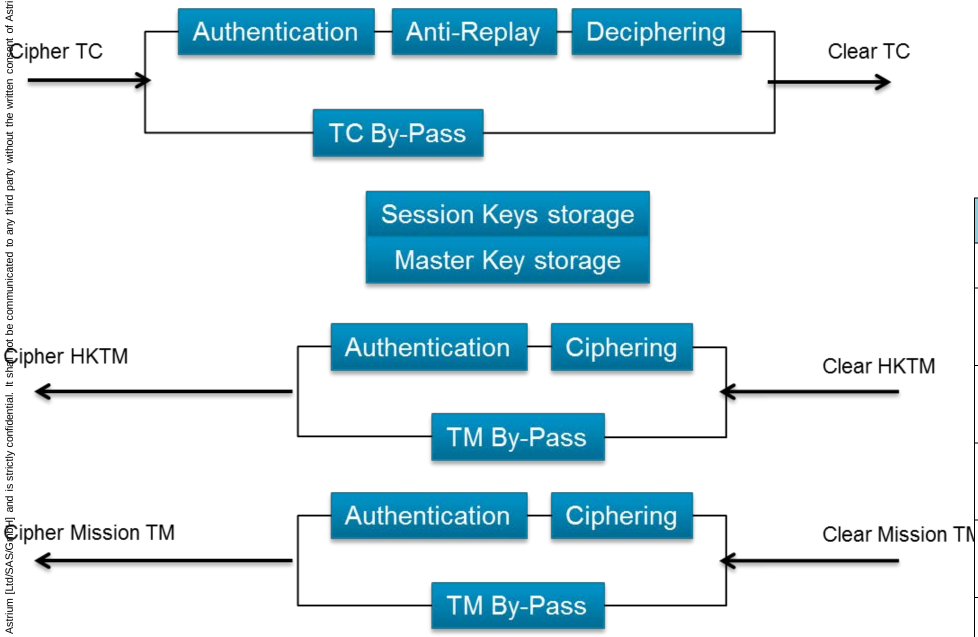
Functional level - Approach

- FMEA-like approach, with effect split on mission (dependability, safety) and security

- Effect before, and after, FDIR

Security Unit	Failure mode	Potential causes	Mission, Safety	Security
Ciphering of HK TM				
	No ciphering of HK TM	Unwanted by-pass commanding Security Unit internal failure	No net effect	Protected HK TM may be left vulnerable to unauthorized viewing
	Erroneous ciphering of HK TM	Security Unit internal failure Corrupted Session Keys	Loss (corruption) of HK TM	Confidentiality is not endangered since data is corrupted
Ciphering of Mission TM				
	No ciphering of Mission TM	Unwanted by-pass commanding Security Unit internal failure	No net effect provided mission TM is not lost	Protected Mission TM may be left vulnerable to unauthorized viewing.
	Erroneous ciphering of Mission TM	Security Unit internal failure Corrupted Session Keys	Loss (corruption) of Mission TM Loss of mission if unrecovered.	Confidentiality is not endangered since data is corrupted
(…)				
	(…)			

Functional model, failure modes



Cipher TC

Authentication

Anti-Replay

Deciphering

Clear TC

TC By-Pass

Session Keys storage

Master Key storage

Cipher HKTM

Authentication

Ciphering

Clear HKTM

TM By-Pass

Cipher Mission TM

Authentication

Ciphering

Clear Mission TM

TM By-Pass

Effect at Security Unit Level	Associated Component Failure(s) (TN4-Part 2)
No ciphering of TM	Encryptor main data path failure
Erroneous ciphering of TM	Encryptor main data path failure Oscillator failure
Loss of TC authentication protection / all TCs accepted	Authentication engine – MAC preparation Control interface's authentication engine
Authentication permanently fails	Authentication engine – MAC preparation Control interface's authentication engine
Wrong authentication	Authentication engine – MAC verification Status output authentication engine
Loss of Anti-replay protection / all TCs accepted	Non-volatile Memory
Anti-replay protection refuses all TCs	Non-volatile Memory
No / Erroneous deciphering of TC	Decryptor main data path
Spurious by-pass mode enabling	Bypass switch logic or bypass control SW
By-Pass mode cannot be enabled	Control input interface receiver
Loss or corruption of key(s)	Non-volatile Memory Volatile RAM
Erroneous key selection (stuck or jump)	Control interface's decryptor

FDIR recommendations for security

- Proposition of means in terms of
  - Prevention
    - No by-pass for critical commands (e.g., secure configuration commands)
    - Two-step by-pass command
    - …
  - Detection (ground and board, or ground only)
    - Security Unit health monitoring
    - Periodic checks with invalid MAC, TC replay…
    - …
  - Recovery
    - Preserve mission: command by-pass, ignore MAC, …
    - Preserve security: notion of *secure mode* :
      - E.g., suspend TM, disable by-pass, select new session key automatically, …
    - Preserve both, based on advanced Security Unit redundancy scheme and FDIR, including the ability to physically isolate a failed unit (down to the power distribution level)