The State of CryptoLib

The Open-Source Satellite Cryptography Library

Cody Cutright
System Engineer
TMC Technologies
NASA IV&V's JSTAR Program

GSAW 2024

Who is NASA IV&V?

- NASA Katherine Johnson Independent Verification and Validation Program
  - Ensuring Class A Mission success since 1993
  - Full Development Lifecycle
    - Software Requirements, Design, Code, Test, I&T
  - Located in West Virginia

- JSTAR Team
  - The Jon McBride Software Testing & Research Laboratory
    - Named for WV's first astronaut
  - Specialized subset of IV&V Facility
  - Experts in modeling, simulation, and digital twins of NASA missions

JIST

ARRISTOTLE

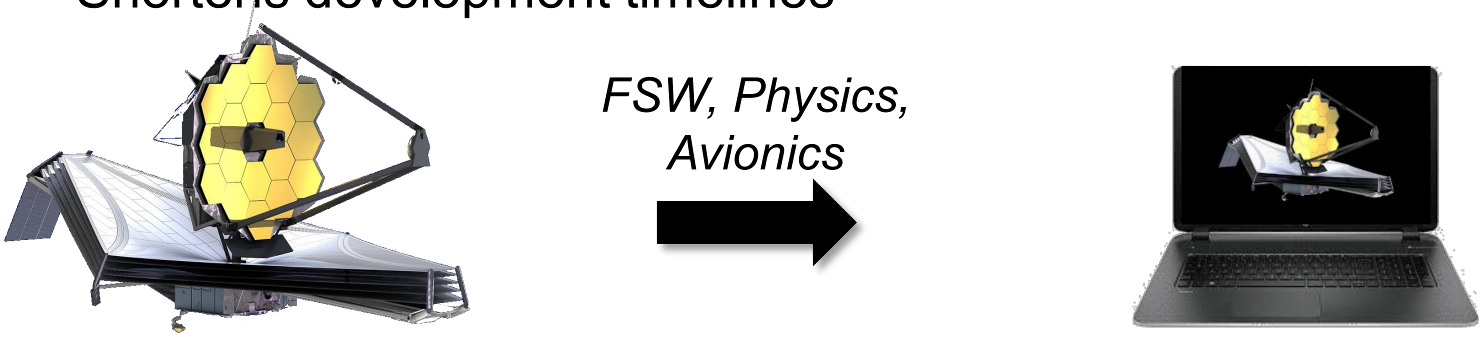
ADVANCED RISK REDUCTION INTEGRATED SOFTWARE TEST & OPERATIONS TRI-PROGRAM LIGHTWEIGHT ENVIRONMENT

WISP

RST DIGITAL TWIN

What Does JSTAR Do?

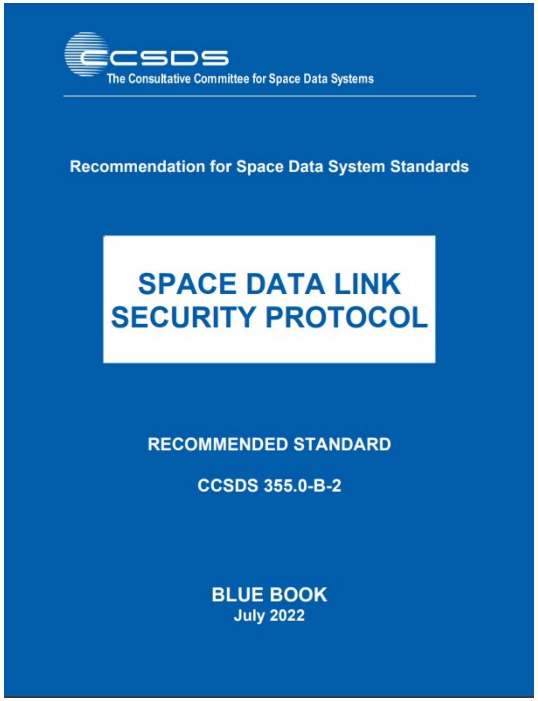
- Experts in software only simulation of hardware
- Experts in hardware emulation
- We provide simulated hardware so flight binaries can be executed, **unmodified**, on commodity hardware
  - This enables unique, challenging, and otherwise difficult to create test cases
  - Virtual testbeds allow development to proceed without immediate access to a flatsat
  - In some cases, code can essentially be finalized prior to hardware integration
  - Shortens development timelines



What is CryptoLib?

CryptoLib is an open-source implementation of the CCSDS Space Data Link Systems (SDLS) and SDLS-Extended Procedures (SDLS-EP) Standards, the original implementation by John Lucas at IV&V

- Encrypts & decrypts Transfer Frames:
  - Telecommand (TC)
  - Telemetry (TM)
  - Advanced Orbiting System (AOS)
- Open-sourced library
- C Based
- Utilizes LibgCrypt by default
- Support for WolfSSL
- Support for JPL’s KMC (Bouncy castle)



CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

SPACE DATA LINK SECURITY PROTOCOL

RECOMMENDED STANDARD

CCSDS 355.0-B-2

BLUE BOOK
July 2022

*From ccsds.org

https://github.com/nasa/CryptoLib

Why CryptoLib?

- CubeSats - Missions are not exempt based on size or budget
- CCSDS Protocols already in use
  - Telecommand and telemetry protocols are in wide usage
  - Additional standards provide guidance for encryption / authentication
- An unencrypted command link has implications for other missions
- No PhD Required – Minimal Encryption Knowledge necessary!
- Open-Source - No black box, transparent handling of the process
- No Vendor Lock in
- Save $$$
- NASA-STD-1006A

NASA-STD-1006A

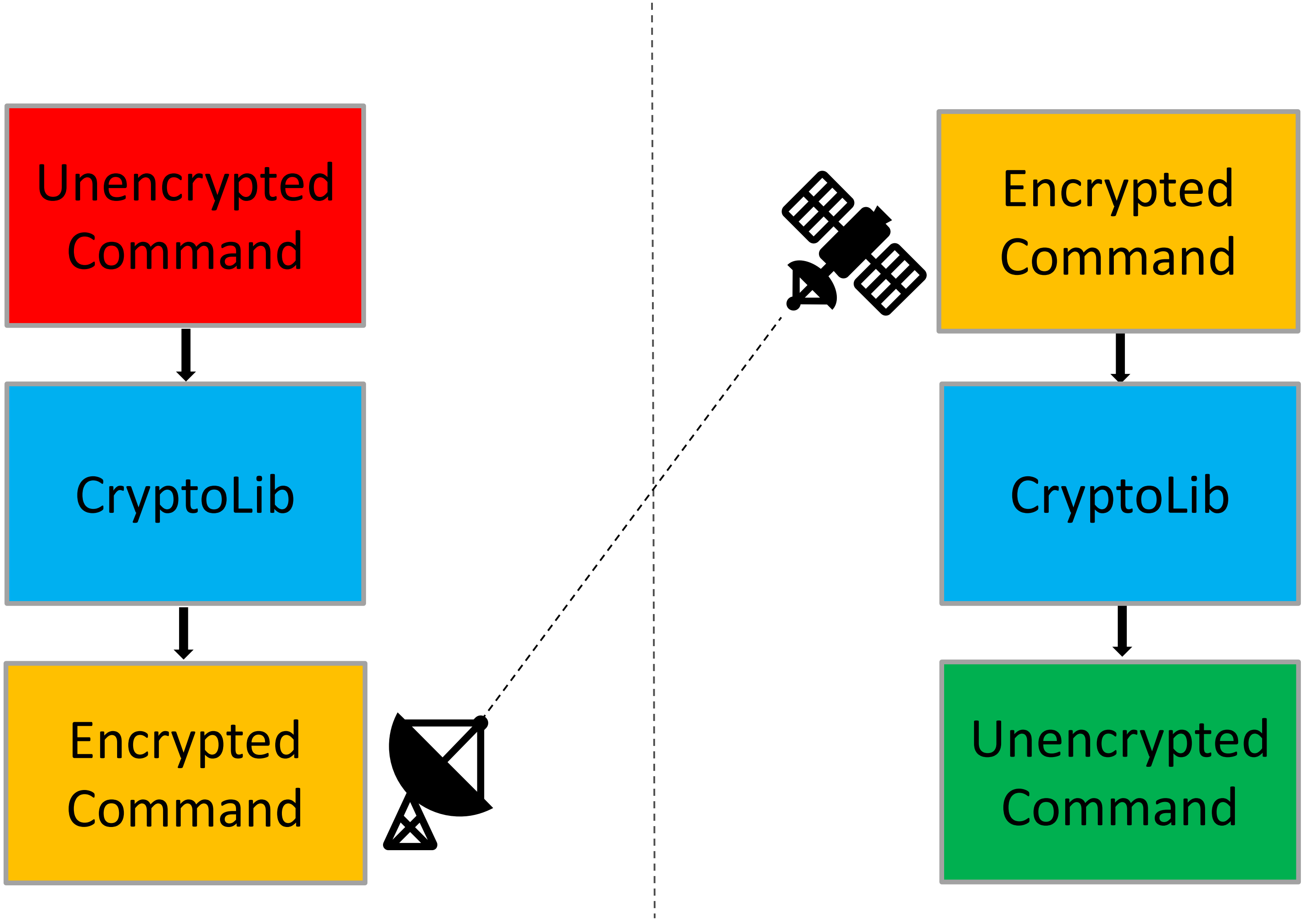
"Programs/projects shall protect the command stack with encryption that meets or exceeds the Federal Information Processing Standard (FIPS) 140, Security Requirements for Cryptographic Modules, Level 1. "

[…]

iii. Deep space missions may choose to limit controls applied to the space link if certain controls (e.g., encryption and authentication) pose significant burden to operability or mission success, and if the threat to the space link is low.

iv. Category 3/Class C or Class D missions may authenticate without encryption if they have no propulsion.

High Level Overview



Unencrypted Command

CryptoLib

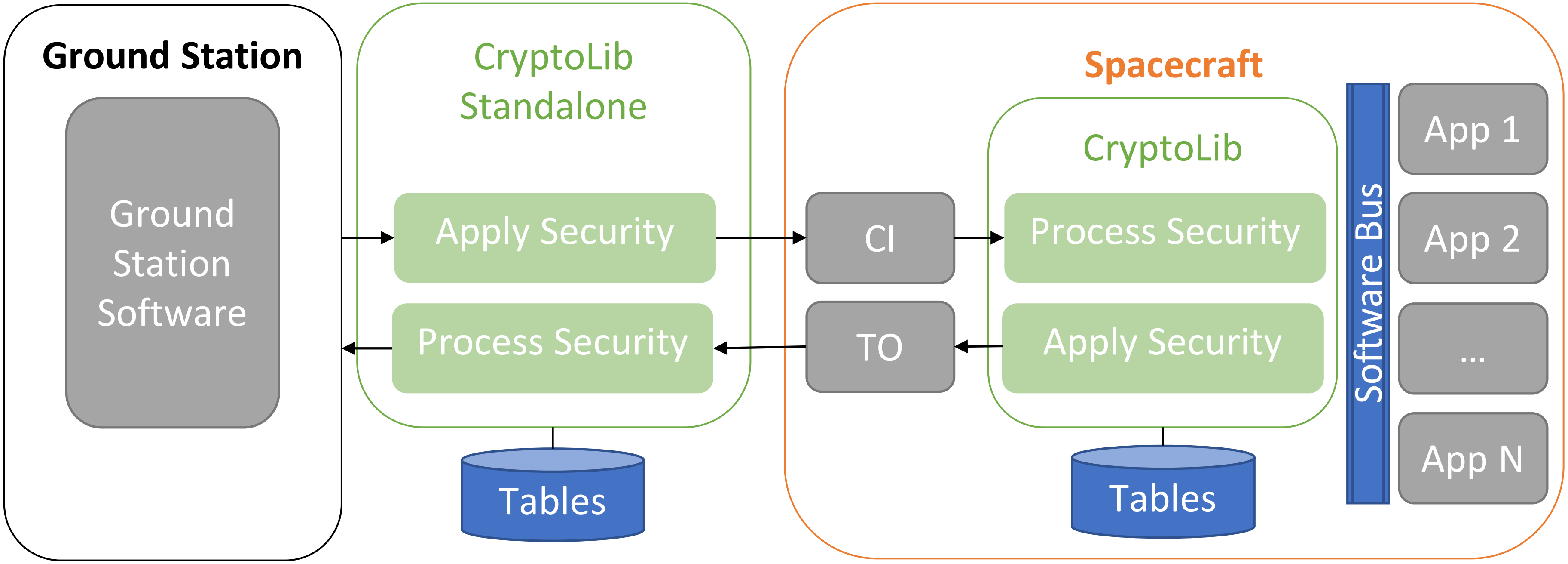
Encrypted Command

Encrypted Command

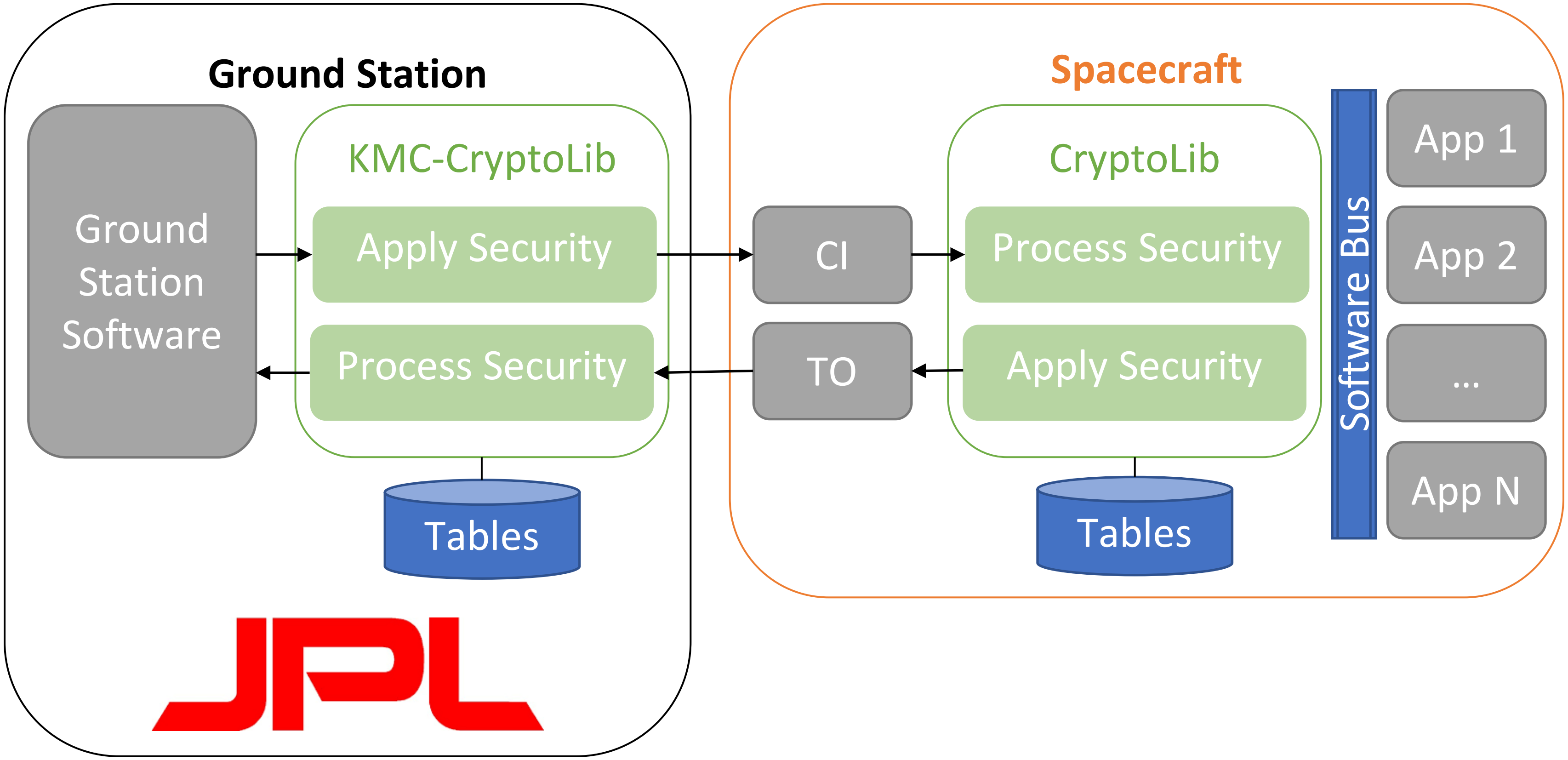
CryptoLib

Unencrypted Command

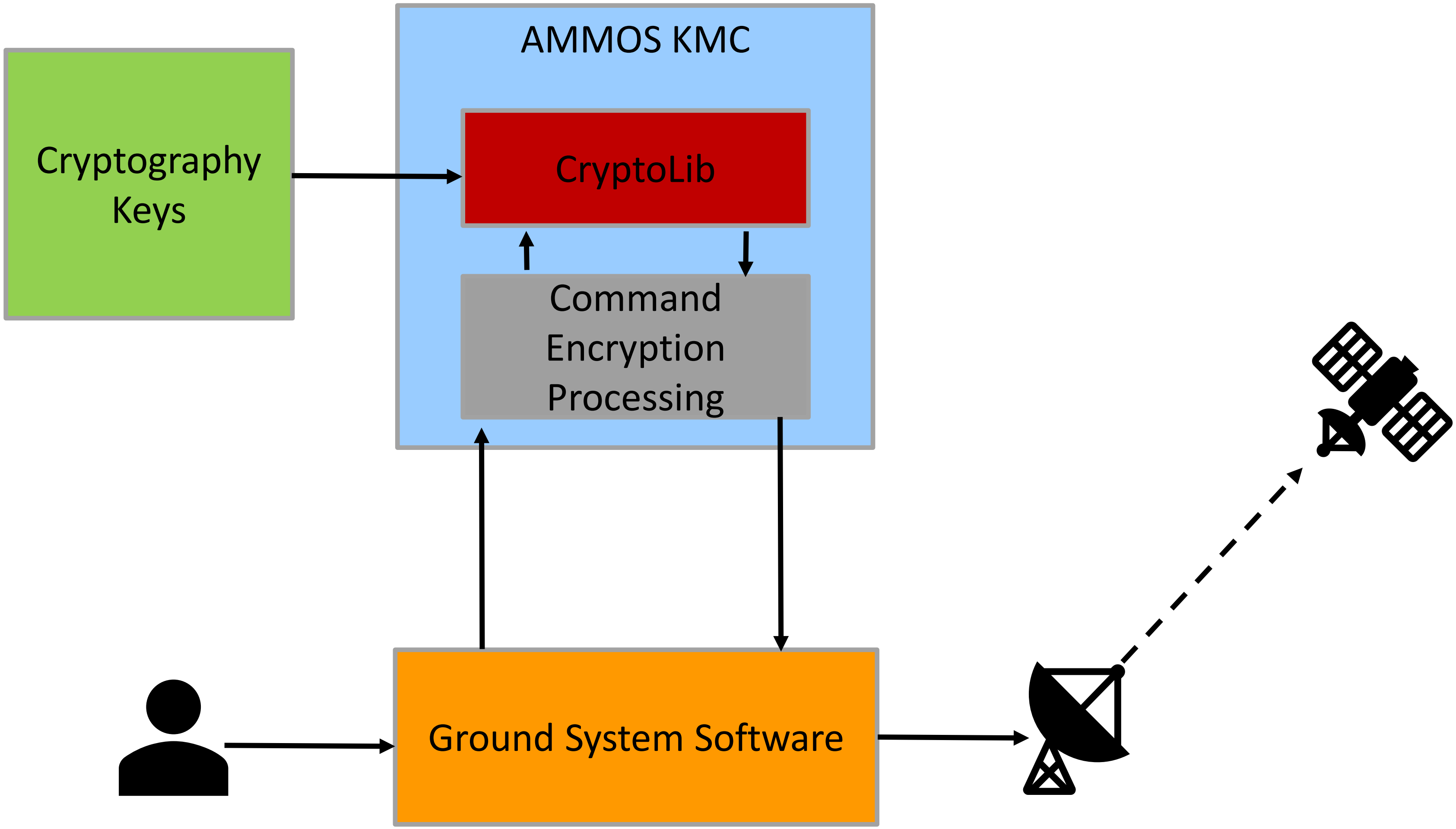
CryptoLib: Standalone Configuration



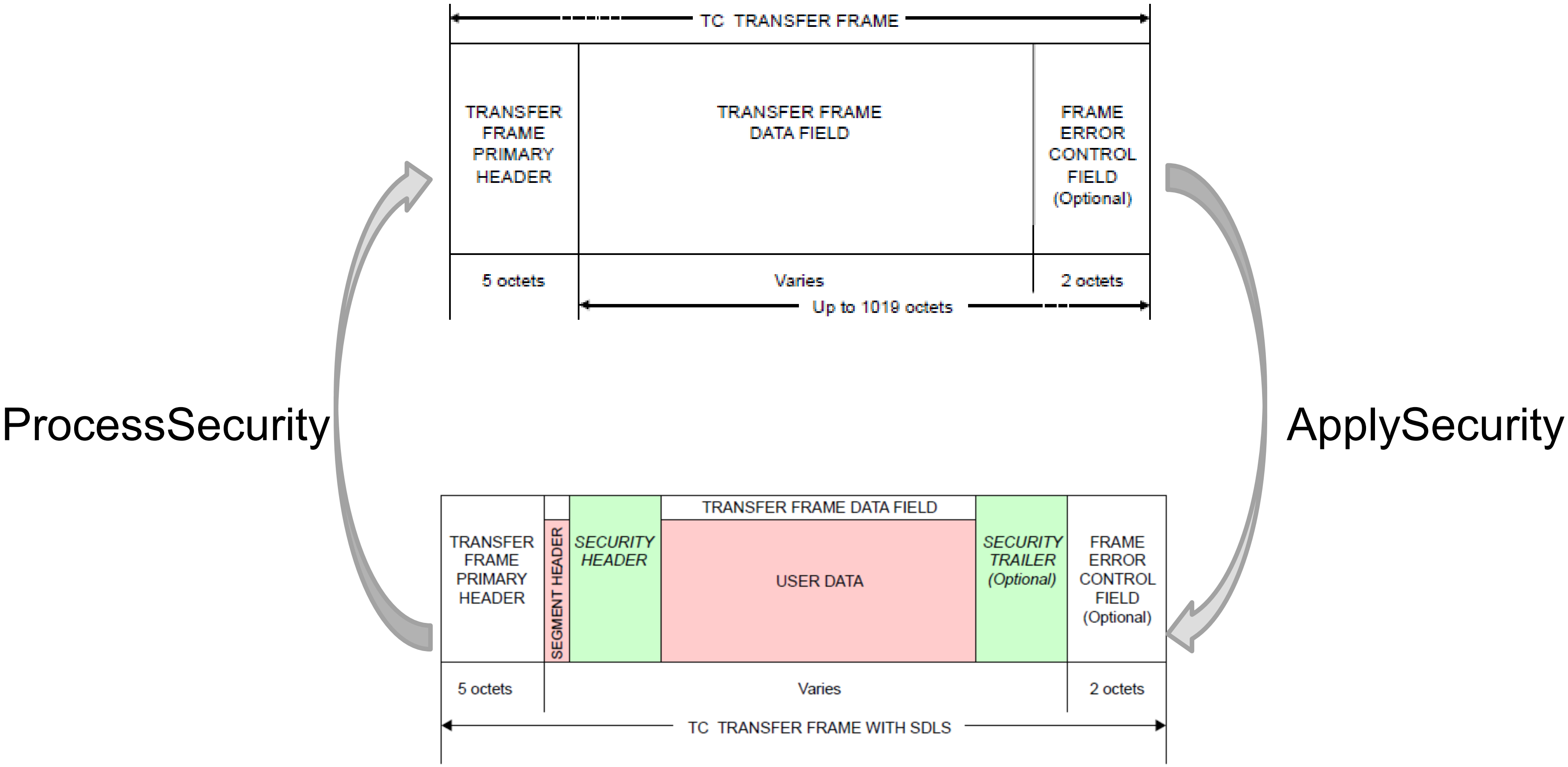
CryptoLib: KMC Integration



KMC: External Keystore



Telecommand Transfer Frames



TC TRANSFER FRAME

TRANSFER FRAME PRIMARY HEADER	TRANSFER FRAME DATA FIELD	FRAME ERROR CONTROL FIELD (Optional)
5 octets	Varies	2 octets

Up to 1019 octets

ProcessSecurity

ApplySecurity

TRANSFER FRAME PRIMARY HEADER	SEGMENT HEADER	*SECURITY HEADER*	TRANSFER FRAME DATA FIELD USER DATA	*SECURITY TRAILER (Optional)*	FRAME ERROR CONTROL FIELD (Optional)
5 octets	Varies				2 octets

TC TRANSFER FRAME WITH SDLS

*From CCSDS 232.0-B-4

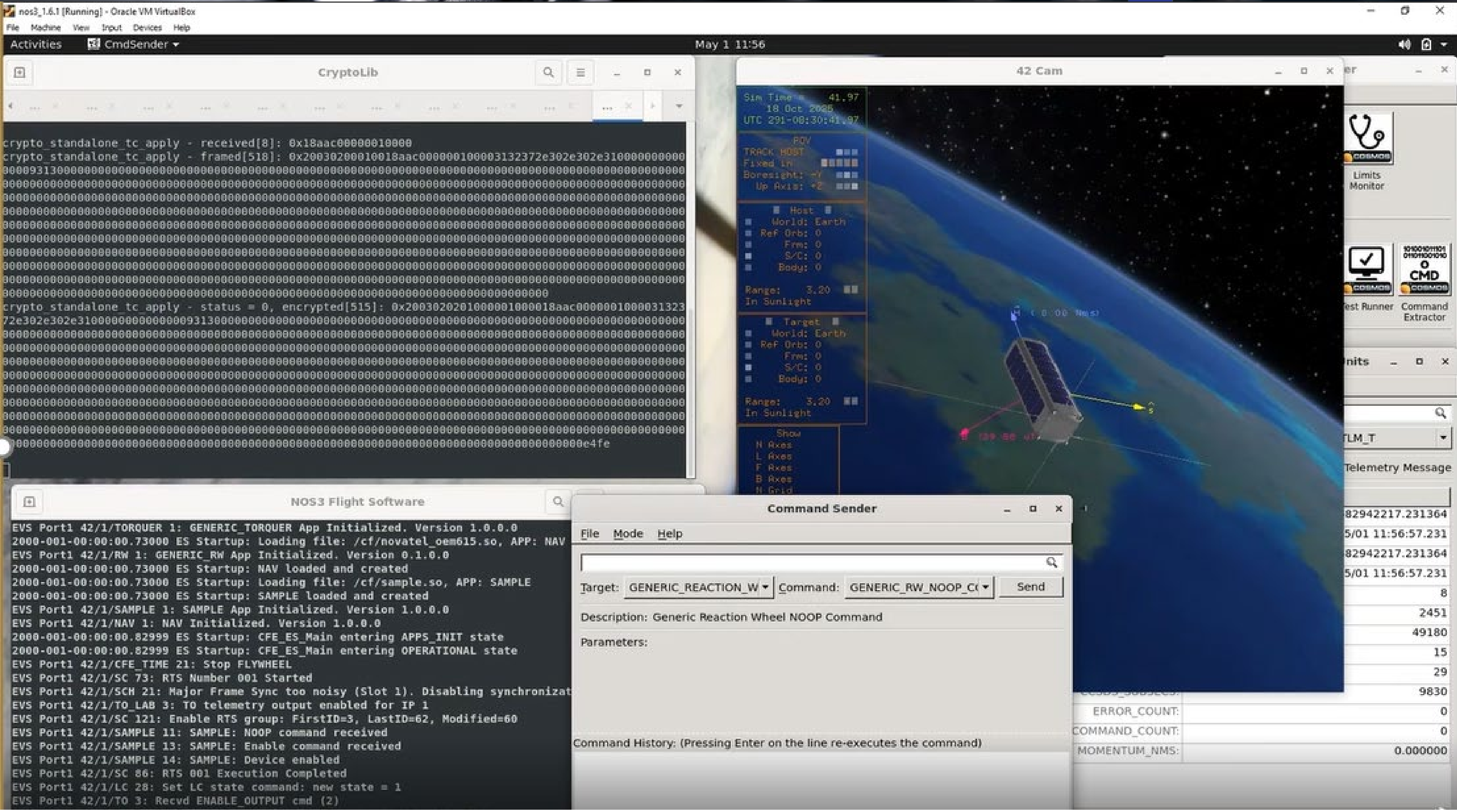
Current Status

- Functional Telecommand Uplink
  - CCSDS Telecommand (TC) Transfer Frames
- Functional Telemetry Downlinks
  - CCSDS Telemetry (TM) Frames
  - CCSDS Advanced Orbiting Systems (AOS) Frames
- Security Options
  - Authentication
  - Encryption
  - Authenticated Encryption
  - Plaintext
- Implemented algorithms
  - AES GCM 256
  - AES CBC 256
  - AEC CCM
  - SHA 256/512 Hashing

Current Status

- Standalone Module
- Integration with NASA’s core Flight System
- Integration with JPL’s Key Management & Cryptography
- Integration with NASA Operational Small Satellite Simulator (NOS^3)
  - https://github.com/nasa/nos3
- Supports WolfSSL
- Interoperability tested with JPL's Lunar Trailblazer and SunRISE

NOS³



```
crypto_standalone_tc_apply - received[8]: 0x18aac00000010000
crypto_standalone_tc_apply - framed[518]: 0x20030200010018aac000000100003132372e302e302e310000000000
0000931300000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
crypto_standalone_tc_apply - status = 0, encrypted[515]: 0x2003020201000001000018aac0000001000031323
72e302e302e31000000000000009313000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000e4fe
```

```
EVS Port1 42/1/TORQUER 1: GENERIC_TORQUER App Initialized. Version 1.0.0.0
2000-001-00:00:00.73000 ES Startup: Loading file: /cf/novatel_oem615.so, APP: NAV
EVS Port1 42/1/RW 1: GENERIC_RW App Initialized. Version 0.1.0.0
2000-001-00:00:00.73000 ES Startup: NAV loaded and created
2000-001-00:00:00.73000 ES Startup: Loading file: /cf/sample.so, APP: SAMPLE
2000-001-00:00:00.73000 ES Startup: SAMPLE loaded and created
EVS Port1 42/1/SAMPLE 1: SAMPLE App Initialized. Version 1.0.0.0
EVS Port1 42/1/NAV 1: NAV Initialized. Version 1.0.0.0
2000-001-00:00:00.82999 ES Startup: CFE_ES_Main entering APPS_INIT state
2000-001-00:00:00.82999 ES Startup: CFE_ES_Main entering OPERATIONAL state
EVS Port1 42/1/CFE_TIME 21: Stop FLYWHEEL
EVS Port1 42/1/SC 73: RTS Number 001 Started
EVS Port1 42/1/SCH 21: Major Frame Sync too noisy (Slot 1). Disabling synchronizat
EVS Port1 42/1/TO_LAB 3: TO telemetry output enabled for IP 1
EVS Port1 42/1/SC 121: Enable RTS group: FirstID=3, LastID=62, Modified=60
EVS Port1 42/1/SAMPLE 11: SAMPLE: NOOP command received
EVS Port1 42/1/SAMPLE 13: SAMPLE: Enable command received
EVS Port1 42/1/SAMPLE 14: SAMPLE: Device enabled
EVS Port1 42/1/SC 86: RTS 001 Execution Completed
EVS Port1 42/1/LC 28: Set LC state command: new state = 1
EVS Port1 42/1/TO 3: Recvd ENABLE_OUTPUT cmd (2)
```

```
Sim Time =      41.97
   18 Oct 2025
UTC 291-08:30:41.97

        FOV
TRACK HOST
Fixed in
Boresight: -Y
  Up Axis: +Z

     Host
   World: Earth
 Ref Orb: 0
     Frm: 0
     S/C: 0
    Body: 0

Range:    3.20
In Sunlight

    Target
   World: Earth
 Ref Orb: 0
     Frm: 0
     S/C: 0
    Body: 0

Range:    3.20
In Sunlight

     Show
  N Axes
  L Axes
  F Axes
  B Axes
  N Grid
```

Command Sender

File Mode Help

Target: GENERIC_REACTION_W ▾ Command: GENERIC_RW_NOOP_C(▾ Send

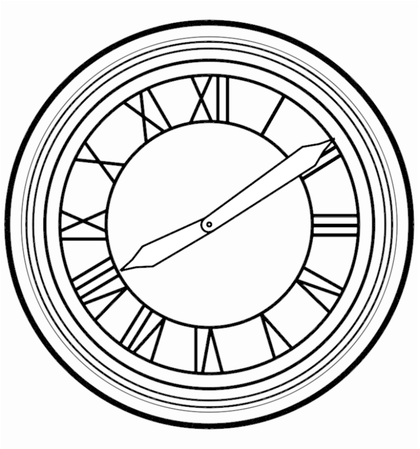
Description: Generic Reaction Wheel NOOP Command

Parameters:

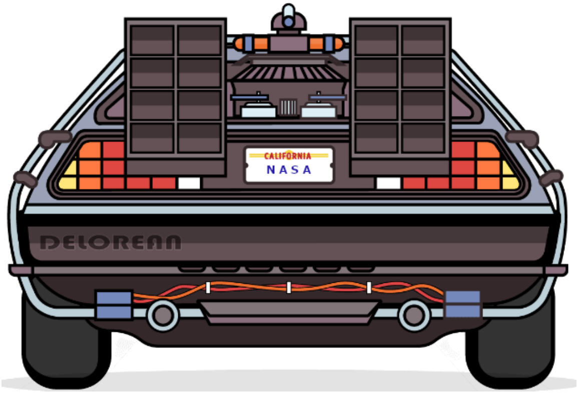
Command History: (Pressing Enter on the line re-executes the command)

	82942217.231364
	5/01 11:56:57.231
	82942217.231364
	5/01 11:56:57.231
	8
	2451
	49180
	15
	29
	9830
ERROR_COUNT:	0
COMMAND_COUNT:	0
MOMENTUM_NMS:	0.000000

The Future



- Increasing Unit Test Coverage to 100%
- Reducing Cyclomatic Complexity
- Addition of Extended Procedures
  - Over The Air Rekeying (OTAR)
  - Key lifecycle management
  - Security Association Management
- Integration with additional ground systems
  - ASIST / FEDS
  - ITOS
  - COSMOS



Major Contributors

- Robert Brown, Systems Engineer, NASA IV&V, Robert.J.Brown@nasa.gov
- David "Cody" Cutright, Systems Engineer, NASA IV&V, David.C.Cutright@nasa.gov
- John Lucas, Systems Engineer, NASA IV&V, John.P.Lucas@nasa.gov
- Michael Pajevski, AMMOS Architect, NASA JPL, Michael.J.Pajevski@jpl.nasa.gov
- Scott Zemerick, Chief Systems Engineer, NASA IV&V, Scott.A.Zemerick@nasa.gov
- Ibraheem Y Saleh, Software Engineer

Questions / Answers