Dissecting ARID

Implementing and Evaluating Security Solutions on Open-Source Drones

*College of Science and Engineering (CSE), Hamad Bin Khalifa University (HBKU), Doha, Qatar

+Eindhoven University of Technology, Faculty of Mathematics and Computer Science, Eindhoven, Netherlands

Pietro Tedeschi*

Savio Sciancalepore+, Roberto Di Pietro*

Learning from Authoritative Security Experiment Results (LASER) 2021 | ACSAC 2021

December 6-10, 2021, Online

Agenda

Motivations
Security and Privacy Issues of UAVs

ARID Phases
Remote Identification *aka* RemoteID vs ARID

Implementation Details
What we did in terms of programming

Performance Assessment
ARID and its performance assessment

Conclusion and Future Work
What we offered and what are the future ARID extensions

ARID

Motivations

Worldwide Drone Incidents

- Police and air traffic control intervene after drone spotted at Newcastle – Stadiums - Newcastle, United Kingdom - October 8, 2021
- Criminals Use Drones to Drop 5 Liters of Flammable Liquid - Law Enforcement/First Responders - October 1, 2021
- Drone crashes into Leaning Tower of Pisa - Private/Non-Corporate - Pisa, Italy - September 28, 2021
- Drone spies on private home - Private/Non-Corporate - Albringhausen, Germany - July 11, 2021 …

SOURCE: HTTPS://WWW.DEDRONE.COM/

FAA Remote IDentification Rule

- Enhance accountability of Unmanned Aerial Vehicles(UAVs) operations
- It forces all UAVs operators to broadcast messages reporting their identity, location (GPS position), and information about ground station
- *RemoteID* does not specify how to generate such identifiers, nor provide guidelines to operators for their design
- Effective on the 21st of April 2021, and UAV operators need to comply with this rule from September 2022

Federal Aviation Administration
REMOTE ID RULE

UAV must periodically broadcast messages containing at least the following information

Unique ID	Drone Latitude, Longitude, Altitude, Speed	GCS Latitude, Longitude, Altitude	Timestamp	Emergency Status

Motivations


dronelife
News Products Industries Enthusiasts Regulations
When Will Remote ID Go Into Effect? FAA Announces Date
Posted By: Miriam McNabb on: March 13, 2021

Nextgov
TRENDING // THE FIRST 100 DAYS // THE HACK // 5G // SPONSORED: APPLIED INTELLIGENCE // SPONSORED:
CYBERSECURITY EMERGING TECH ARTIFICIAL INTELLIGENCE IT MODERN
FAA Delays Drone Remote ID Tracking, 'Operations Over People' Rules

AINonline
BIZAV AIR TRANSPORT DEFENSE EVENTS
READ AIN'S 2021 NBAA-BACE
BUSINESS AVIATION
NBAA: Remote ID Drone Rule Raises Privacy Concerns
by Kerry Lynch - January 22, 2021, 10:19 AM

Privacy nightmare? FAA's drone tracking rules have big consequences
New rules require broadcast of information that could compromise delivery customers' privacy.
By Greg Nichols for Robotics | January 4, 2021 | Topic: Robotics
RELATED

Cybersecurity Alert: FAA Releases New Rules Concerning Drone Use

ARID - PHASES

1) Registration Phase

2) On-Line Phase

3) Reporting Phase

ARID: Anonymous Remote IDentification

1 *Registration Phase*

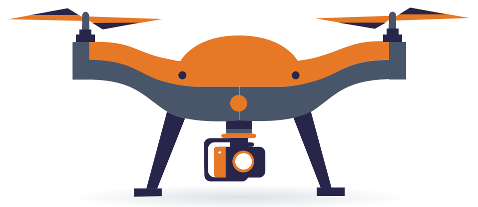
Register an UAV with the Authority, in a way to enable unique identification and receives the cryptography materials necessary to run ARID.

2 *Online Phase*

UAV generates and emits RemoteID-compliant messages, enabling operators to identify their locations, while still preserving UAV anonymity.

3 *Reporting Phase*

Triggered by a Critical Infrastructure (CI) operator, when it detects an invasion of the protected area by an UAV, CI reports the attack to the Auth.



Why we implemented ARID

ARID is implemented because:

- To evidence the compliance with the flying constrained devices (UAVs as the new “Flying IoT”)
- To evaluate the Performance on limited battery lifetime devices with a minimum impact!
- Of releasing open-source code to enhancing the impact of ARID, demonstrating its deployability to improving the quality of the provided security services in real-world UAV systems
- To demonstrate and evaluate that it is possible develop and integrate a real (security & privacy) solution even on commercial UAVs

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

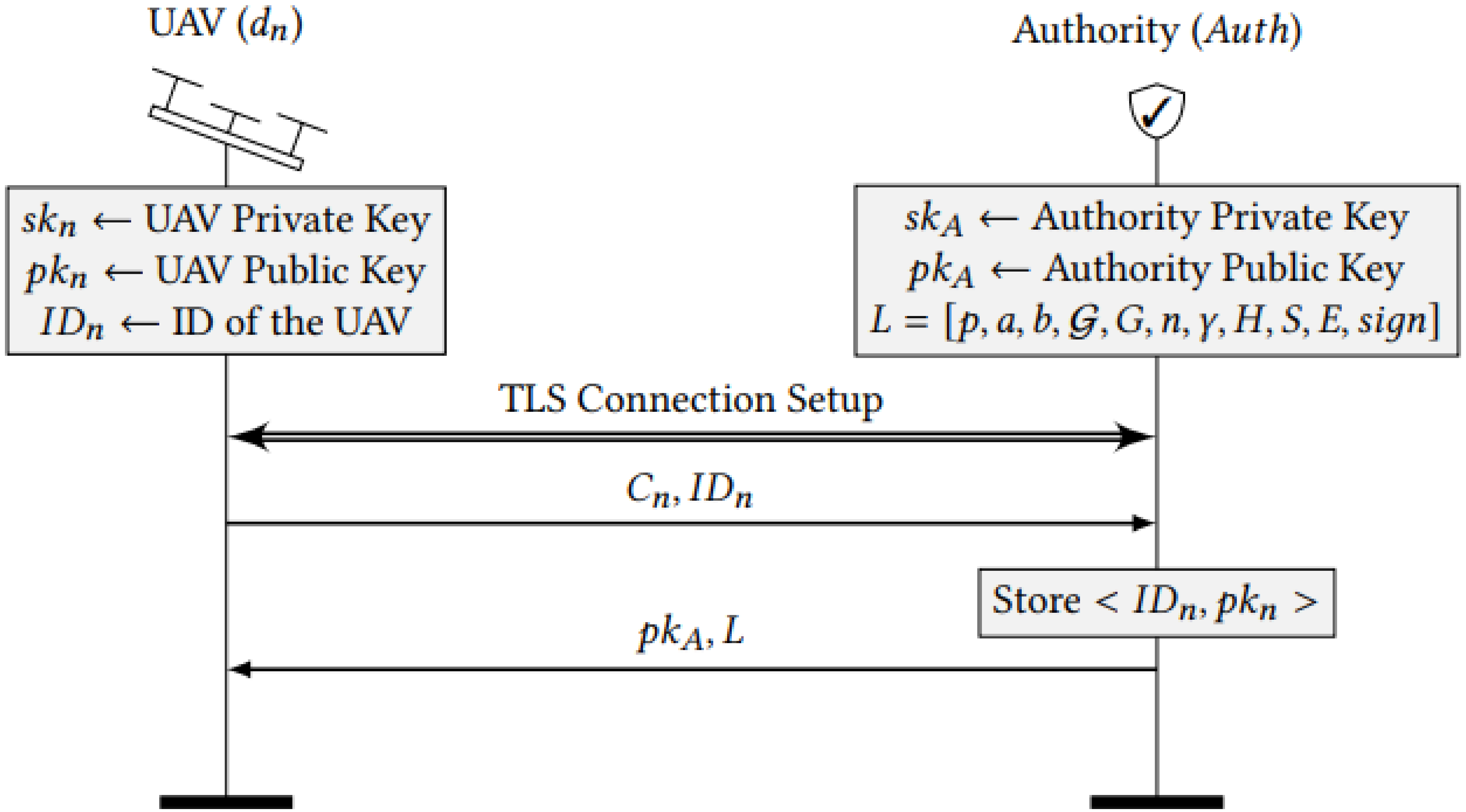
OpenSSL
Cryptography and SSL/TLS Toolkit

GitHub

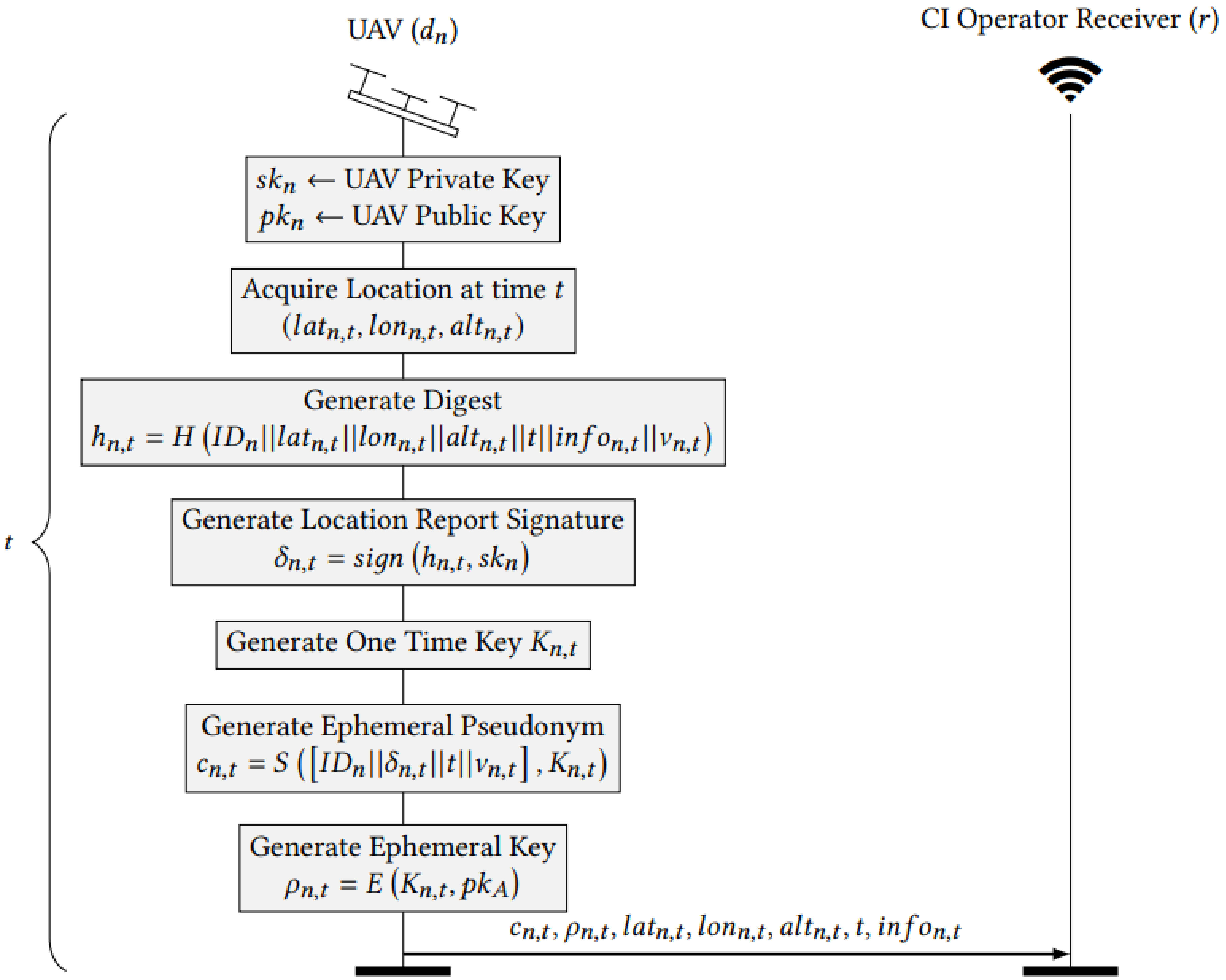
ARID AT WORK!

Implementation Details

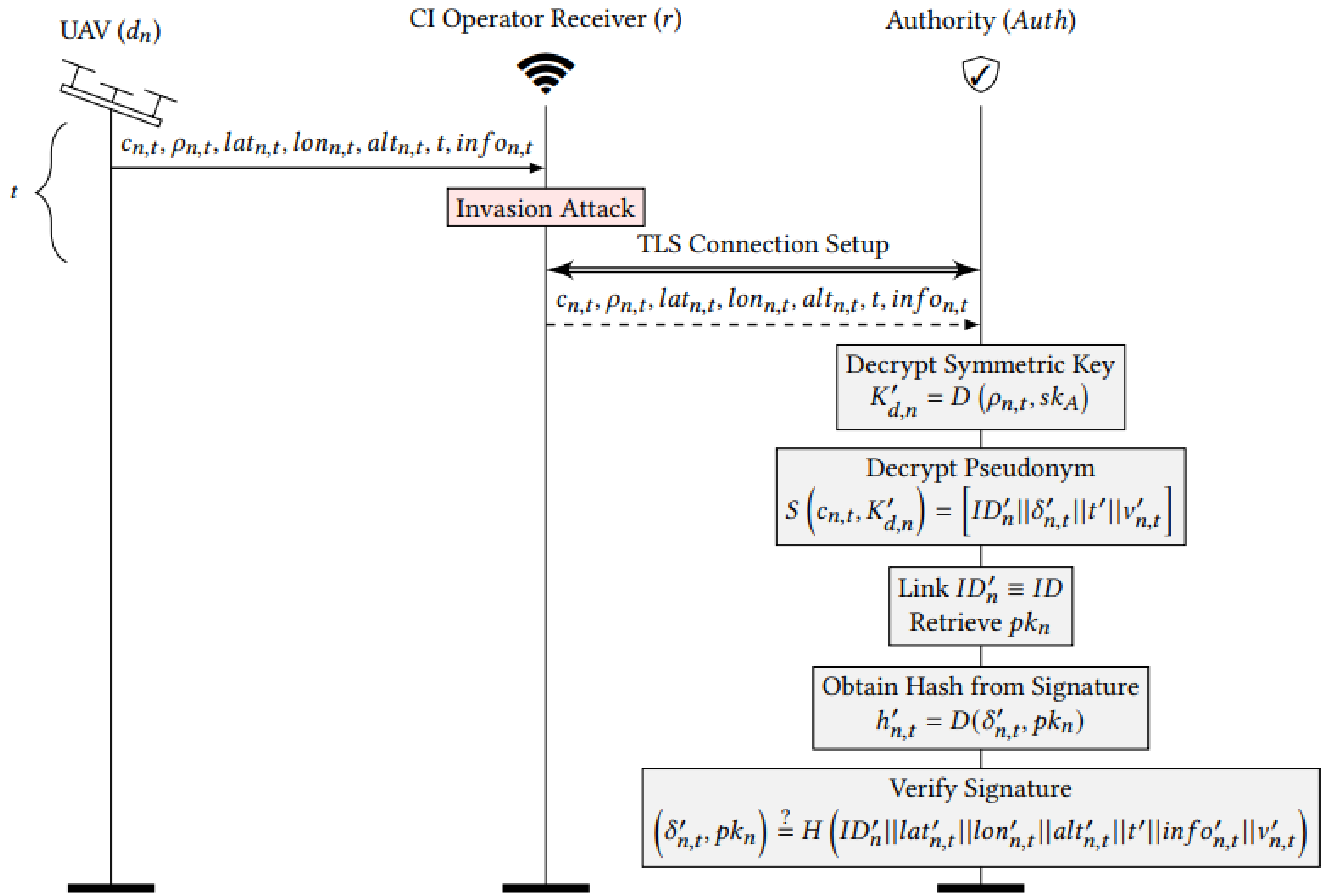
Registration Phase



Online Phase



Reporting Phase



Drone Features – 3DR Solo Drone

- CPU i.MX6 Solo - 1.00GHz ARM Cortex A9
- 7,948.00 MB ROM, 512MB RAM
- Cryptographic Acceleration and Assurance Module
- WiFi Module supports IEEE 802.11b
- 3DR Poky OS 1.5.1 – Linux based OS

HW/SW Requirements

Laptop
A laptop equipped with a GNU/Linux distro

Drone
It can execute ELF 32-bit LSB (i.e. OS Linux)
It supports MAVLink 1.0/2.0
Ardupilot/PX4 flight controller

NIC in Monitor Mode (optional)
ALFA Card AWUS036NH/AWUS036ACH

Software Defined Radio (optional)
- HackRF One
- gps-sdr sim (to generate GPS baseband signal data strams for indoor tests)

GNU Arm Embedded Toolchain
sudo apt-get install gcc-arm-linux-gnueabihf g++-arm-linux-gnueabihf

Sniffing Tool(s)
sudo apt-get install -y tcpdump wireshark

OpenSSL Library 1.0, MAVLink 1.0
Compile OpenSSL 1.0 (libssl.so.1.0.0) for ARM and install the library on the UAV. **Compile it on your PC and install it manually via SSH.**

GCC, G++ cross compilers
- sudo apt-get install libc6-armel-cross libc6-dev-armel-cross binutils-arm-linux-gnueabi libncurses5-dev build-essential bison flex libssl-dev bc

Cryptography Details

Hashing Function

SHA-256

Asymmetric Enc/Dec Algorithm

El-Gamal

Signature Algorithm

Elliptic Curve Digital Signature Algorithm (ECDSA)

Symmetric Enc/Dec Algorithm

AES-128

Elliptic Curve Cryptography

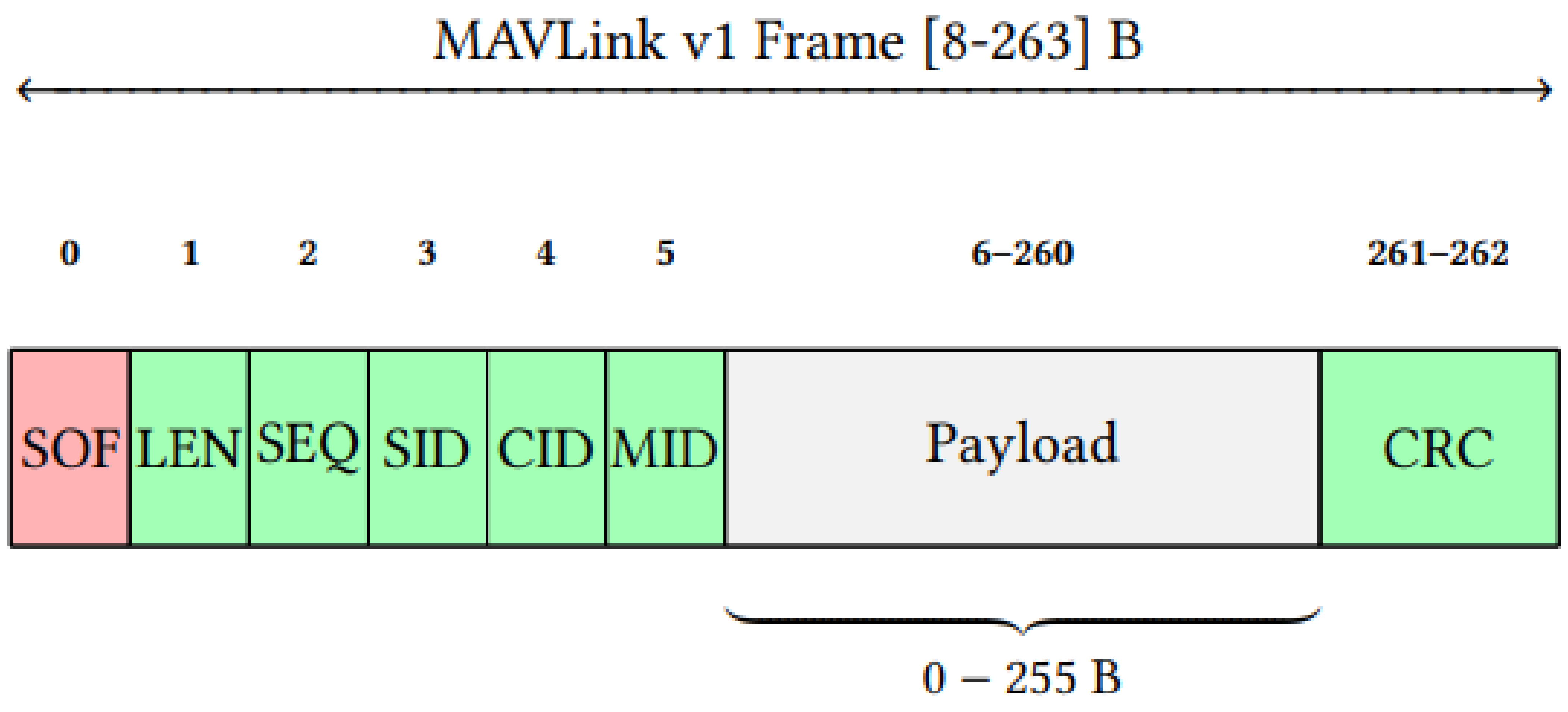
secp160r1, secp192k1, secp224k1, secp256k1

PseudoRandom Number Generator

Read 2048 bits from /dev/urandom

RAND_load_file("/dev/urandom", 256)

Implementation Details



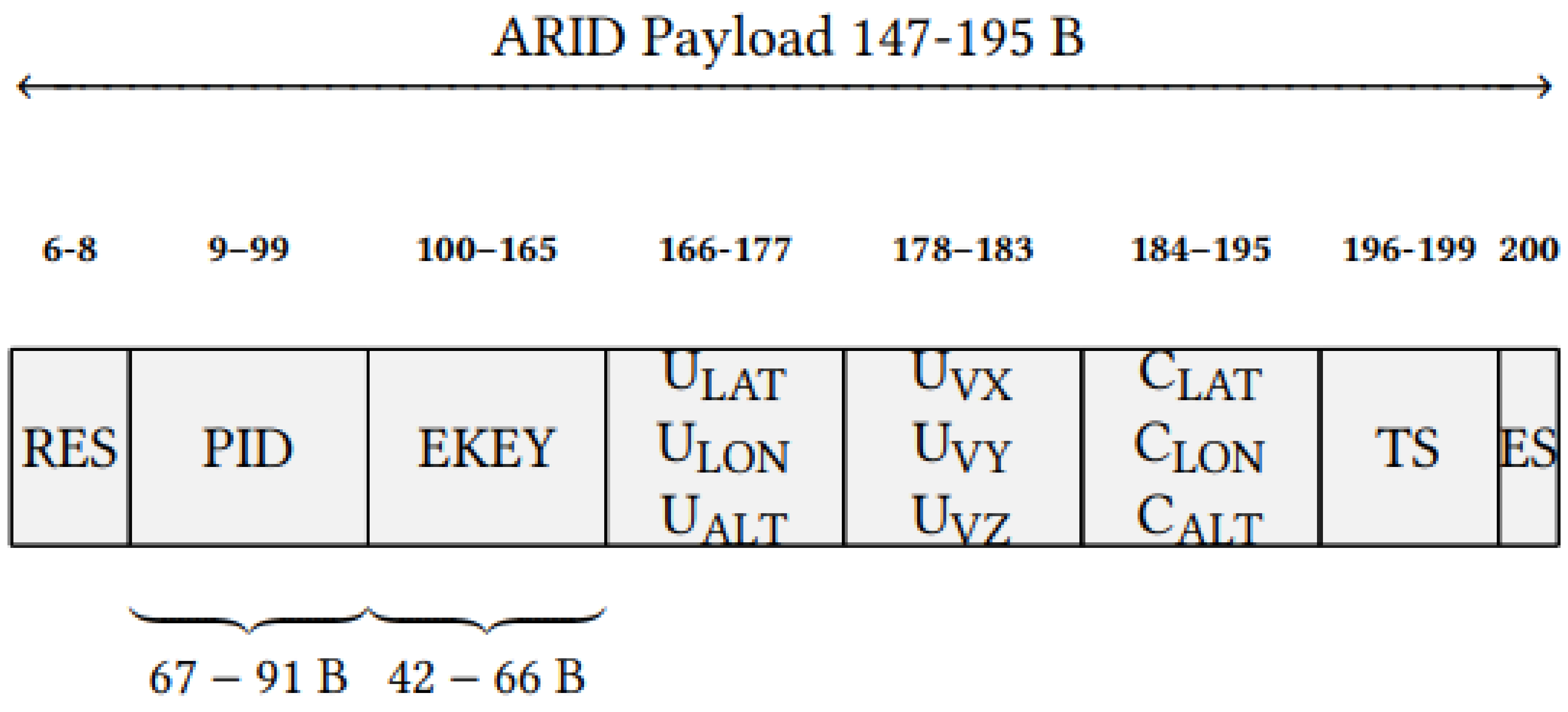


Table 1: MAVLink Frame and ARID Payload Notation.

Acronym	**Content/Size**	**Description**
SOF	0xFE	MAVLink 1.0. Start of Frame
LEN	0x00-0xFF	Payload Length.
SEQ	0x00-0xFF	Sequence number. The value 0x00 represents the first message.
SID	0x01-0xFF	System Identification number of the UAV.
CID	0x00-0xFF	System Identification number of the component that is transmitting the message.
MID	0x00-0xFF	Message Type Identification number. Set to 0XDE for ARID.
Payload	0-255 B	ARID message.
CRC	2 B	Checksum for integrity check.
	ARID Payload	
RES	3 B	Reserved Bytes (e.g. Network, System and Component ID).
PID	67-91 B	UAV Pseudonym $c_{n,t}$ (size based on selected curve).
EKEY	42-66 B	Encrypted one-time key $\rho_{n,t}$ (size based on selected curve).
U_{LAT},U_{LON},U_{ALT}	12 B	UAV Latitude, Longitude, and Altitude (4 bytes each).
U_{VX},U_{VY},U_{VZ}	6 B	UAV Speed x, y and z axis (2 bytes each).
C_{LAT},C_{LON},C_{ALT}	12 B	Latitude, Longitude, and Altitude of the Ground Station (4 bytes each).
TS	4 B	Message Timestamp.
ES	0x00-0xFF	UAV Emergency Status.

ARID MAVLink Custom Message

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

ARID ID MAVLink Message Definition 1.0 (common.xml): 0xDE

```
<message id="222" name="ARID_PROTOCOL">
            <description>File transfer message</description>
            <field type="uint8_t" name="target_network">Network ID (0 for broadcast)</field>
            <field type="uint8_t" name="target_system">System ID (0 for broadcast)</field>
            <field type="uint8_t" name="target_component">Component ID (0 for broadcast)</field>
            <field type="uint8_t[251]" name="payload">Variable length payload. The length is defined by
the remaining message length when subtracting the header and other fields.  The entire content of this
block is opaque unless you understand any the encoding message_type.  The particular encoding used can
be extension specific and might not always be documented as part of the mavlink specification.</field>
        </message>
```

Main ARID Functions (1)

Function Name	Description
void *ARID_init*();	Initialize OpenSSL and PRNG.
static int *setupKey*(BIGNUM **prv, EC_POINT **pbl, BIGNUM *q, const EC_POINT *G, EC_GROUP *curve, BN_CTX *ctx);	Setup the key material on the device (i.e. load/generate the cryptographic keys)
static void *hex_print*(const void*, size_t);	Print in HEX the content of a variable
void *getPadOneTimeKey*(int, int, EC_GROUP *, BN_CTX *, unsigned char *);	Generate One Time Key and Pad the Key for EC Elgamal Encryption Operation
void *unPadKey*(char *, unsigned int , unsigned char *);	UnPad the Key for EC Elgamal Decryption Operation
int *elgamal_encrypt*(char **, char *, int , const EC_POINT *, EC_GROUP *, BN_CTX *, BIGNUM *);	EC ElGamal Encryption Function
int *elgamal_decrypt*(char **, char *, int , BIGNUM *, EC_GROUP *, BN_CTX *);	EC ElGamal Decryption Function

Main ARID Functions (2)

Function Name	Description
void *encrypt_decrypt*(EVP_CIPHER_CTX *, char *, char *, unsigned char *, unsigned char *, bool);	AES Encryption/Decryption Algorithm
void *clean*(EC_GROUP *g, BN_CTX *c, EVP_MD_CTX *h, EVP_CIPHER_CTX *enc);	Clean the memory and the data structures
double *print_time*(struct timeval *s, struct timeval *e);	Print the time defined in a timeval structure.
void *digest*(EVP_MD_CTX *ctx, const EVP_MD *ptr, char *buffer, unsigned char *dig);	Compute the hash of a string. In this case we adopt SHA-256.
int *initialize_UDP*(int *, struct sockaddr_in *, struct sockaddr_in *, int , int);	Init the socket to receive datagram and support UDP protocol
mavlink_parse_char(chan, buf[i], &msg, &status)	Parse the data stream in order to get a MAVLink message
mavlink_msg_global_position_int_decode(&msg, &gps_position);	Decode data of the message and put it in the variable gps_position

Security Levels and Buffer Lengths

Security Level (bits)	*Description*
80	With the elliptic curve *secp160r1* the total size of the MavLink payload is **147** bytes.
96	With the elliptic curve *secp192k1* the total size of the MavLink payload is **163** bytes.
112	With the elliptic curve *secp224k1* the total size of the MavLink payload is **179** bytes.
128	With the elliptic curve *secp256k1* the total size of the MavLink payload is **195** bytes.

- *Payload Computation with 80 bits Security Level*

- RES (3 bytes) + PID (67 bytes) + EKEY (Enc. Public Key 42 bytes) + UAV GPS (12 bytes) + UAV Speed (6 bytes) + GCS GPS (12 bytes) + TS (4 bytes) + Emergency Code (1 byte)

- **PID = ID (4 bytes) + EC Signature ASN.1 DER (49 bytes) + TS (4 bytes) + Nonce (10 bytes)**

IDE and Compile Syntax

- I used Visual Studio Code on Ubuntu 19.04

- **Compile Syntax**

```
arm-linux-gnueabihf-gcc -I ./mavlink-solo/build/common/ -I /usr/local/openssl/include/ -I /usr/local/include/ -L /usr/local/openssl/lib/ -mcpu=cortex-a9 -o arid arid.c -lcrypto -lpthread -Wl,--no-as-needed -ldl -static
```

No GPS for Indoor Test? No Problem

- Configure GPS-SDR-SIM (https://github.com/osqzss/gps-sdr-sim)
- Configure your Software Defined Radio like the HackRF One with an ANT 500 antenna and a TCXO Clock
- Let's go: **hackrf_transfer -t gpssim.bin -f 1575420000 -s 2600000 -a 1 -x 0**

No drone? No worries!

- Install dronekit-sitl with *pip install dronekit-sitl*

- Start dronekit with the following syntax (for details, please type dronekit-sitl -h): dronekit-sitl plane-3.3.0 --home=-35.363261,149.165230,584,353

- In a second terminal spawn an instance of MAVProxy to forward messages from TCP 127.0.0.1:5760 to other UDP ports like 127.0.0.1:14550 and 127.0.0.1:14551

mavproxy.py --master tcp:127.0.0.1:5760 --sitl 127.0.0.1:5501 --out 127.0.0.1:14550 --out 127.0.0.1:14551

No drone? No worries!

- First connect and start the python script (an easy way)
- Start the ARID protocol on your computer as: **./arid**
- Open *Wireshark* on your network-card interface to see the broadcasted packets.

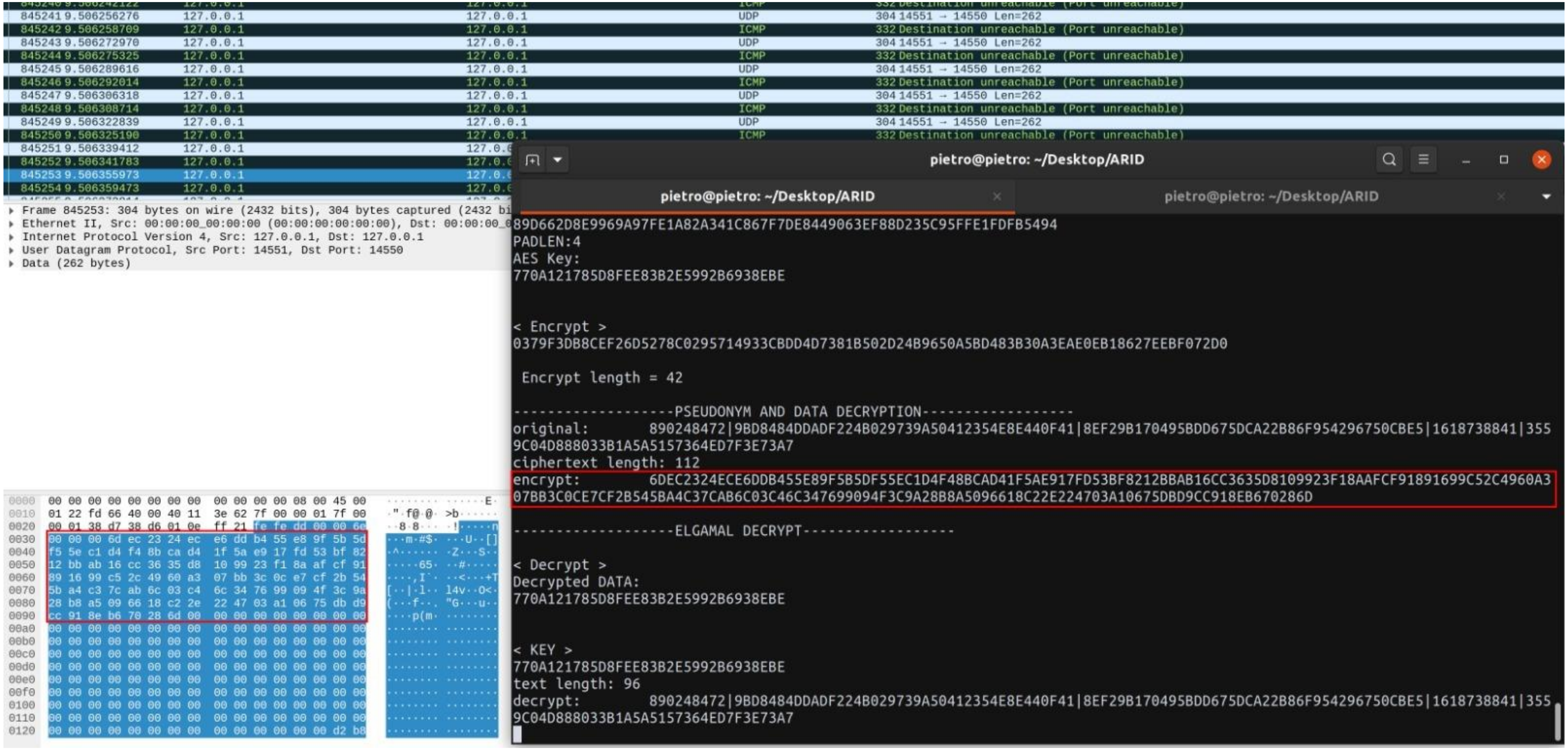
```
print "Start simulator (SITL)"
import dronekit_sitl
sitl = dronekit_sitl.start_default()
connection_string = sitl.connection_string()
from dronekit import connect, VehicleMode
print("Connecting to vehicle on: %s" % (connection_string,))
vehicle = connect(connection_string, wait_ready=True)
print "Get some vehicle attribute values:"
print " GPS: %s" % vehicle.gps_0
print " Battery: %s" % vehicle.battery
print " Last Heartbeat: %s" % vehicle.last_heartbeat
print " Is Armable?: %s" % vehicle.is_armable
print " System status: %s" % vehicle.system_status.state
print " Mode: %s" % vehicle.mode.name    # settable
vehicle.close()
sitl.stop()
print("Completed")
```

Change UAV/Drone MAC Address

- It is easy to change the UAV/Drone MAC address.
- You just need to open an SSH session with the drone and execute the script *change_mac.sh* inside the drone before the flight.
- In this case you will not reveal your legitimate MAC address to potential adversaries.

```
ifconfig wlan0 down
ifconfig wlan0 hw ether 12:34:56:78:12:34
ifconfig wlan0 up
ifconfig
```

Dissecting ARID with WireShark



ARID AT WORK!

Performance Assessment

How did you measure the Time?

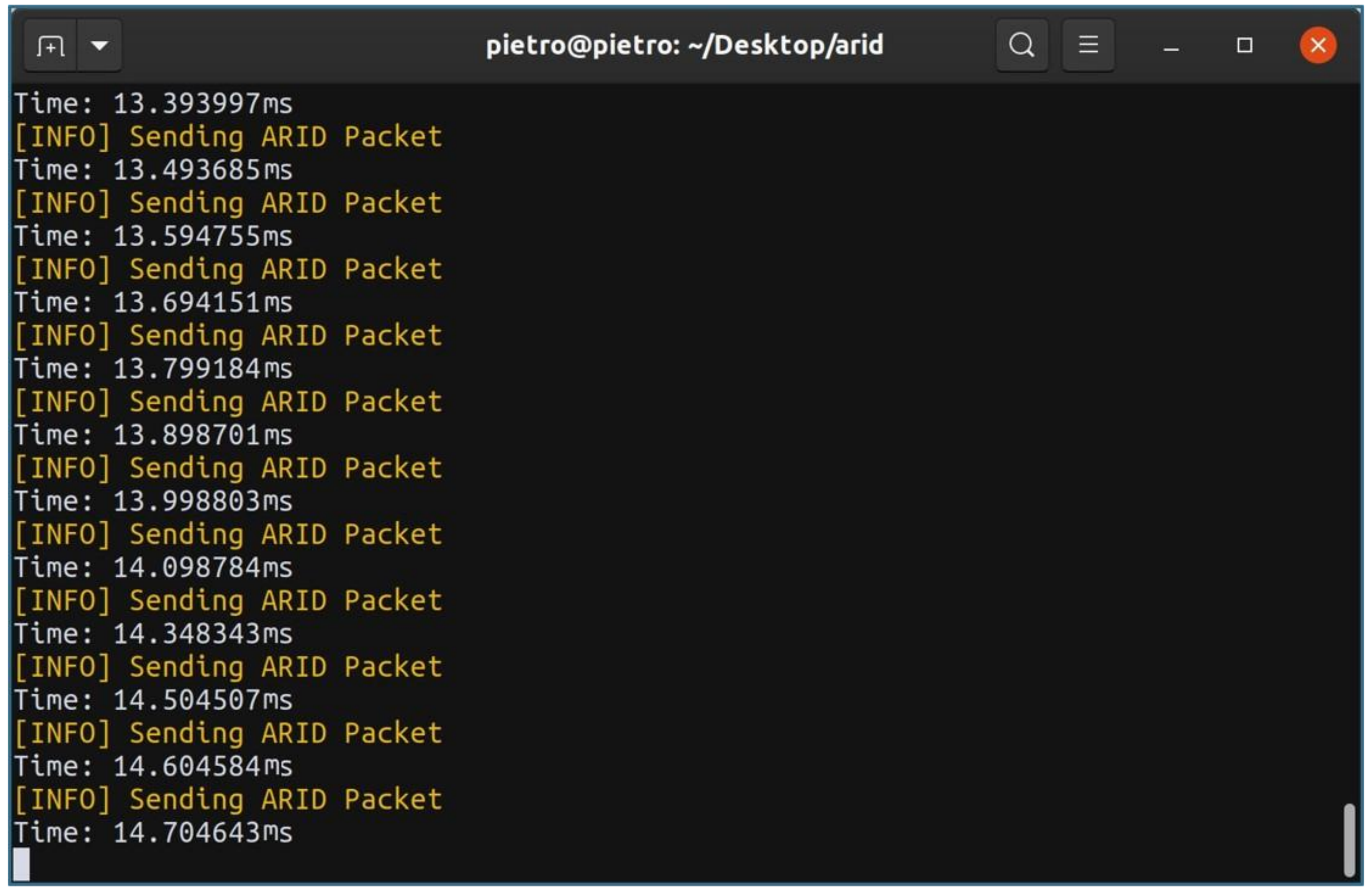
- Measure the time needed to generate and transmit an ARID packet on the 3DR-Solo
- How? Using the *tic-toc* method inside the C script.

- Average time required to execute ARID over 1,000 tests (with 95% confidence intervals)

- Considering the separate contribution of the processing (packet generation, cryptography operations) and radio operations.

- Note that the measured time spans from the GPS location acquisition to the packet delivery (both included).

- Maximum interarrival time T =1 s recommended by the *RemoteID* rule



How did you measure the Energy?

MAVLINK
MICRO AIR VEHICLE COMMUNICATION PROTOCOL

- Telemetry data conveyed by the 3DR-Solo to the remote controller through the MAVLink protocol.

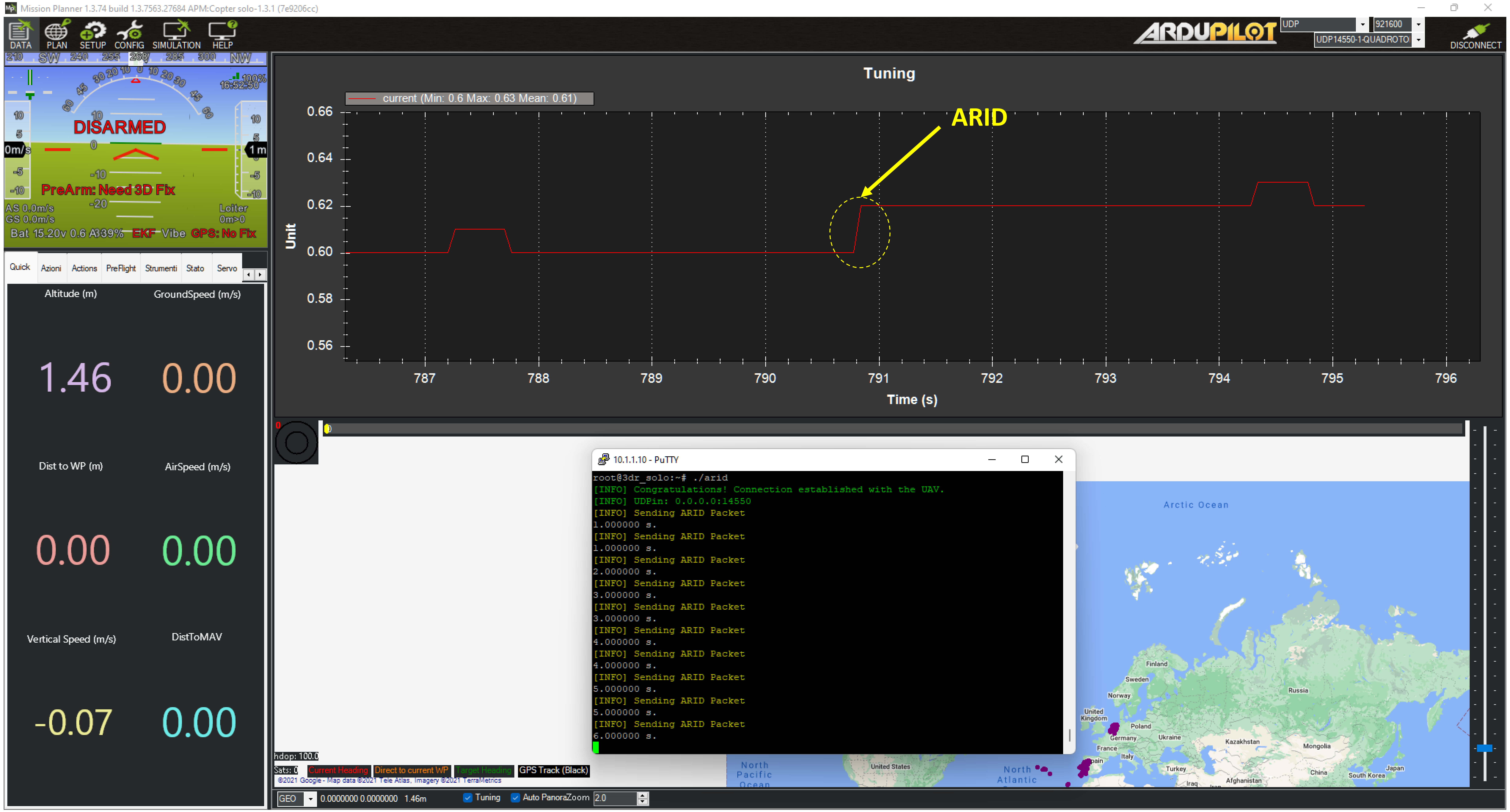
- Measure the difference in the current drained by the drone between: (i) drone at rest; and (ii) during the execution of ARID

- **We computed an average difference of ≈ 20 mA** in the electric current drained by the drone over 1,000 runs

- For the radio operations, the on-board chip of the 3DR-Solo drone belongs to the family AR9300
- 3.3 V, 296.970 mA in TX, 187.879 mA in RX

- IEEE 802.11b, Direct Sequence Spread Spectrum (DSSS) modulation using Differential Binary Phase-Shift Keying (DBPSK)
- Transmission Rate of 1.0 Mbps, 22 MHz channel bandwidth, and a Short Guard Interval of 800 ns.

What about the 20 mA?



How did you measure the Energy?

- Overall Energy Consumption: $\boldsymbol{E[mJ] = V \cdot \int_0^T i(t)dt}$
- V (15.11 V for the UAV's battery and 3.3 V for the radio chip)
- $i(t)$ the instantaneous drained current (20 mA required by ARID on the UAV's battery and 296.970 mA for the radio chip)

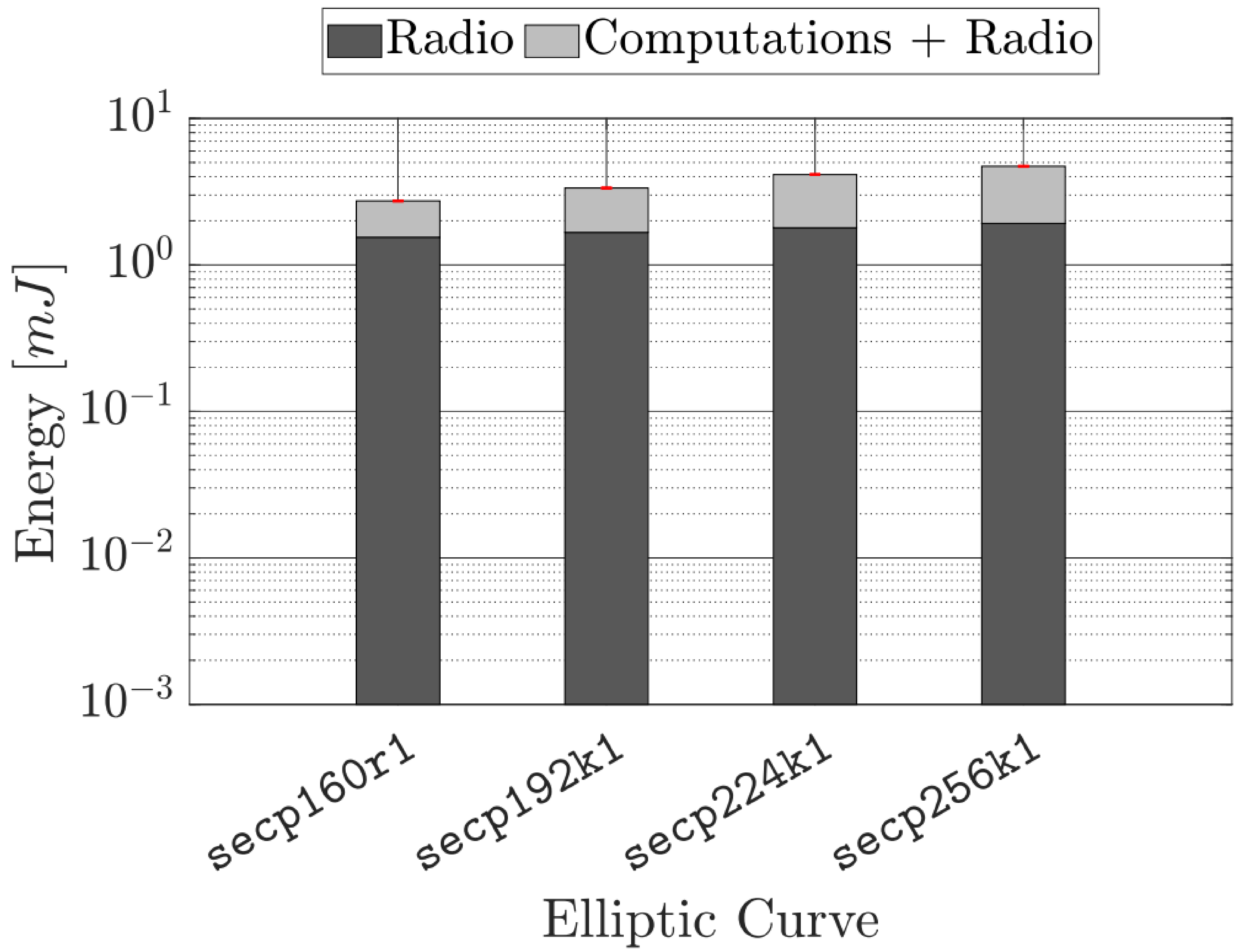
Elliptic Curve	Radio Time (ms)	Comp. Time (ms)	Radio Energy (mJ)	Comp. Energy (mJ)
secp160r1	1.576	3.942 ± 0.0189	1.544	1.191 ± 0.00574
secp192k1	1.704	5.576 ± 0.0286	1.670	1.685 ± 0.00867
secp224k1	1.832	7.781 ± 0.0389	1.795	2.351 ± 0.01176
secp256k1	1.960	9.272 ± 0.0532	1.920	2.802 ± 0.01609

How?

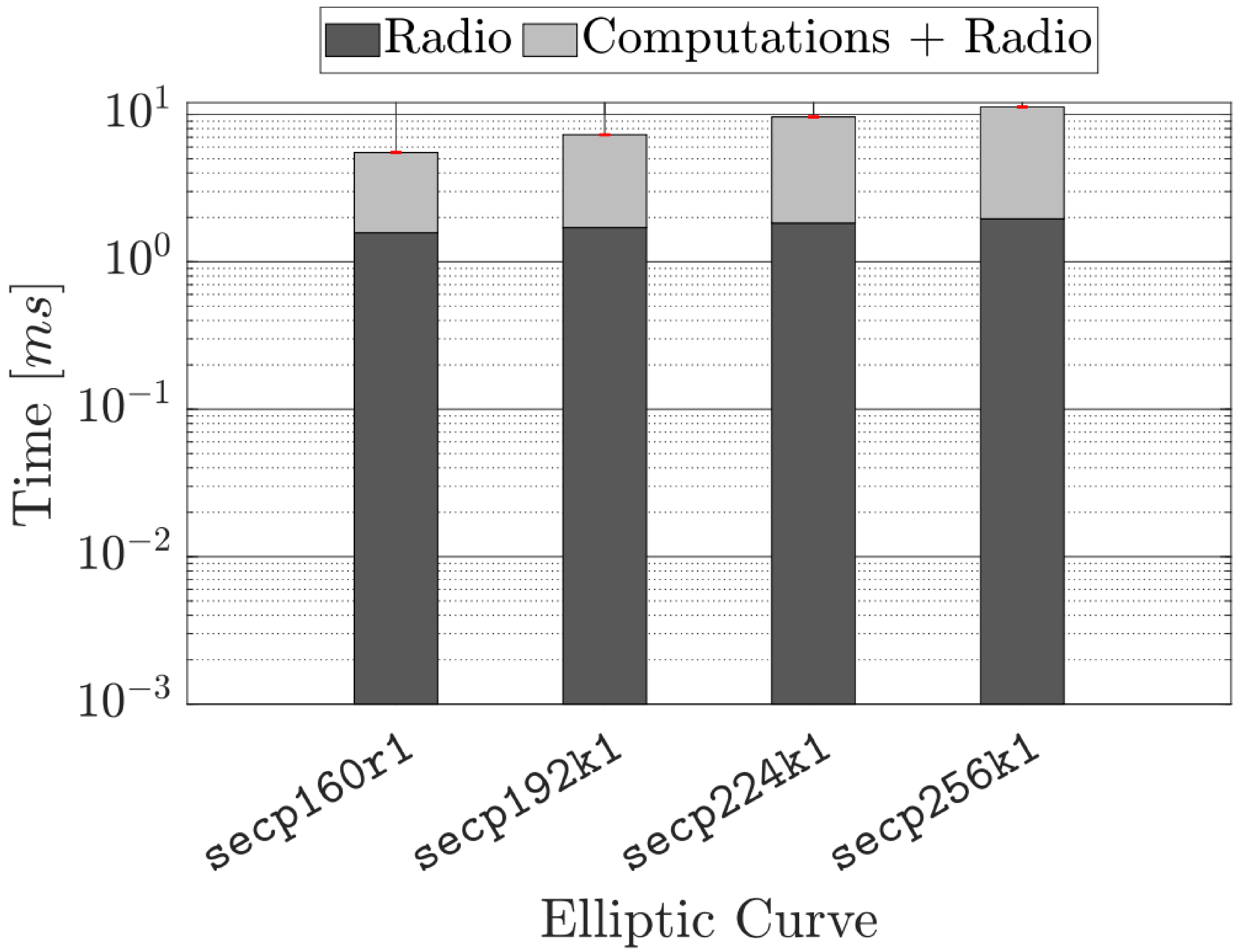
Performance Assessment

Impact of ARID on the battery lifetime. The most energy-consuming configuration of ARID (*secp256k1*) reduces the lifetime of the 3DR-Solo **by only 1.05%** compared to the default (non-anonymous) *RemoteID* configuration, **further demonstrating its limited overhead**.

Energy Consumption



Radio and Computation Time



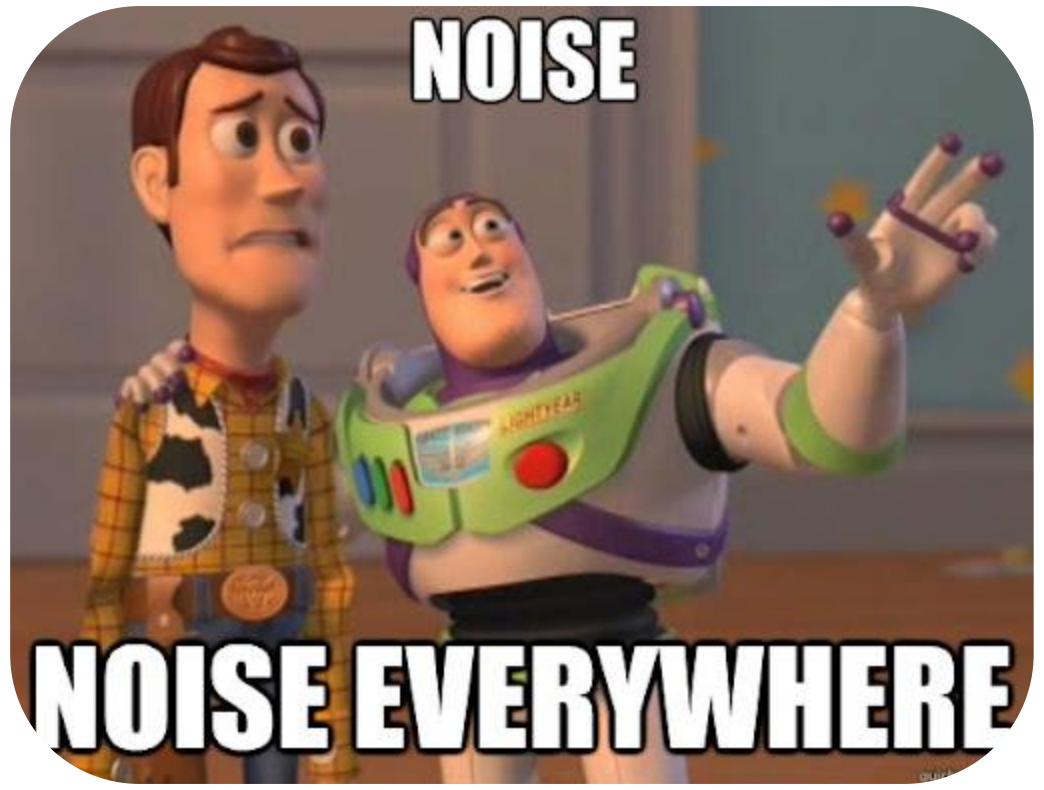
ARID: Impact on Lifetime

1) We executed ARID (*secp256k1*) on the drone with the engines on but without flying. Further we computed the energy consumption!

2) We executed the standard *Remote-ID* protocol (no crypto) on the drone with the engines on but without flying. Further we computed the energy consumption!

3) The differences between 1) and 2) provides you the energy impact of our protocol.

4) Result? **Days and days of noise engine in my head** (even with the headphones)!



ARID: Impact on Lifetime (True Story)

Conclusion and Future Work

- Fully compliant with the latest *RemoteID* regulations by the FAA
- We plan to generalize ARID for other domains in our future work (avionics and maritime networks)
- See you in the main conference ACSAC 21 for more details ☺
- **Proof of Concept released as Open Source @ *https://github.com/pietrotedeschi/arid***

GitHub

THANK YOU !

Any Questions?

PIETRO TEDESCHI, *PhD*
Hamad Bin Khalifa University
e-mail: ptedeschi@hbku.edu.qa
linkedin: https://www.linkedin.com/in/pietrotedeschi/