IN SUPREMÆ DIGNITATIS

1343

Interoperable Maritime Messaging Architecture Using VDES and SECOM Security Standards

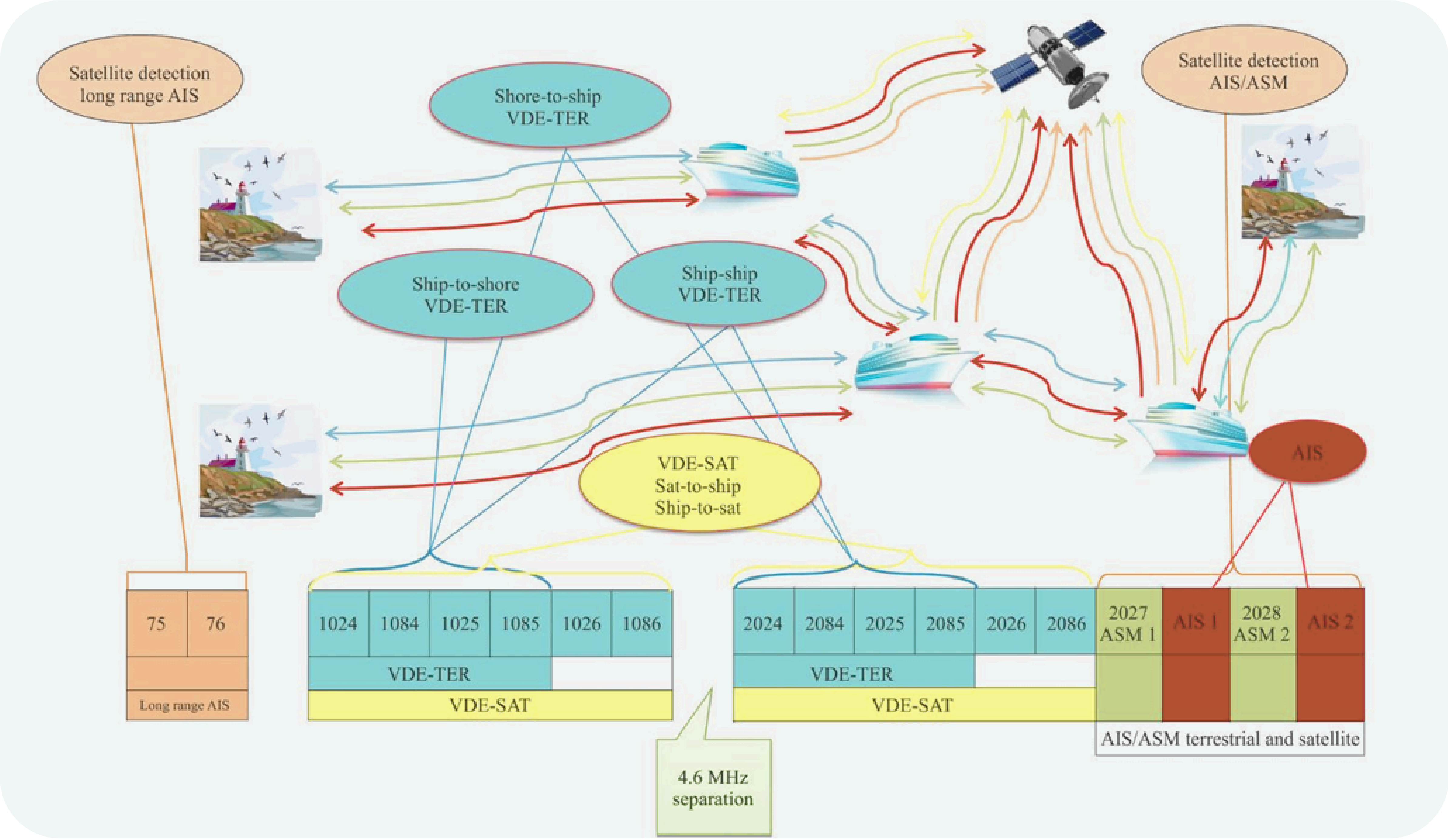
Candidate:
Nicola Lepore

Supervisors:
Prof: Stefano Chessa
Prof: Paolo Pagano

Introduction & Background

WHAT IS VDES & HOW DOES IT WORK?

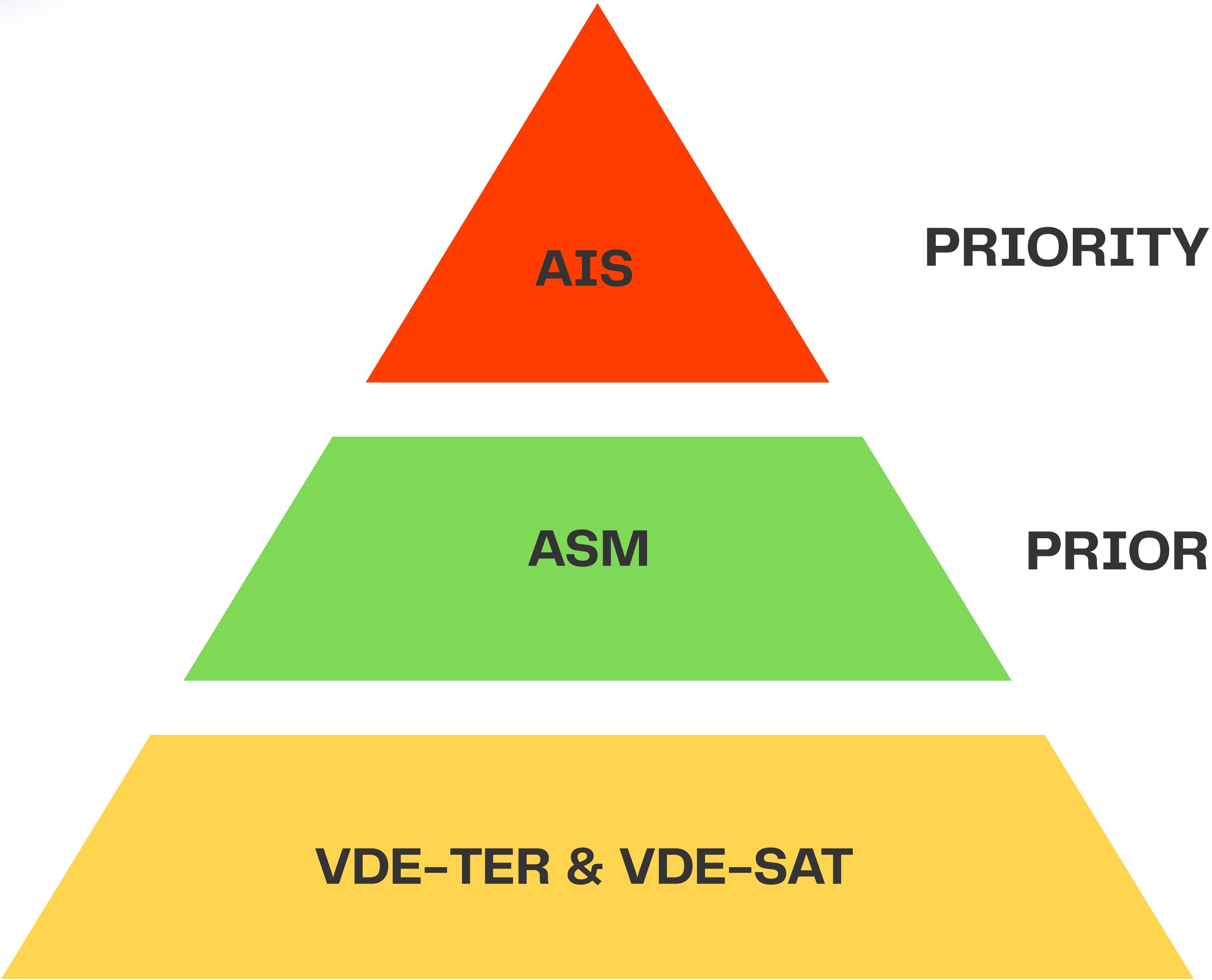
THE VHF DATA EXCHANGE SYSTEM (VDES) IS A RADIO COMMUNICATION SYSTEM OPERATING IN THE MARITIME VHF BAND, DESIGNED FOR THE EXCHANGE OF DIGITAL DATA


Satellite detection long range AIS
Shore-to-ship VDE-TER
Satellite detection AIS/ASM
Ship-to-shore VDE-TER
Ship-ship VDE-TER
AIS
VDE-SAT Sat-to-ship Ship-to-sat
4.6 MHz separation

75	76	1024	1084	1025	1085	1026	1086	2024	2084	2025	2085	2026	2086	2027 ASM 1	AIS 1	2028 ASM 2	AIS 2
		VDE-TER						VDE-TER									
Long range AIS		VDE-SAT						VDE-SAT									
														AIS/ASM terrestrial and satellite			

VDES MESSAGES HIERARCHY

THIS HIERARCHY ENSURES THAT VDES INCREASES COMMUNICATION CAPABILITIES WITHOUT EVER COMPROMISING SAFETY.



AIS

PRIORITY 1 → NAVIGATIONAL SAFETY

ASM

PRIORITY 2 → WEATHER REPORTS OR NOTICES

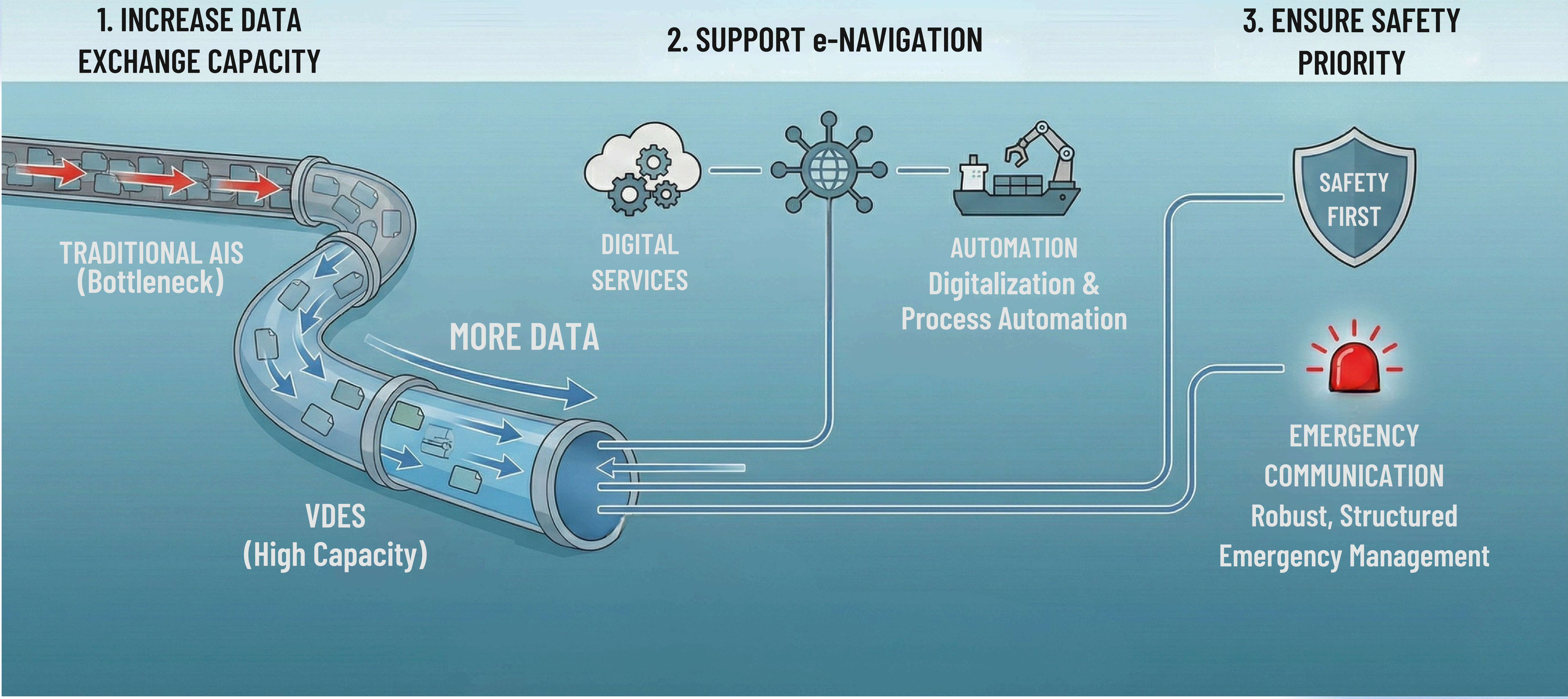
VDE-TER & VDE-SAT

PRIORITY 3 → NON-URGENT DATA

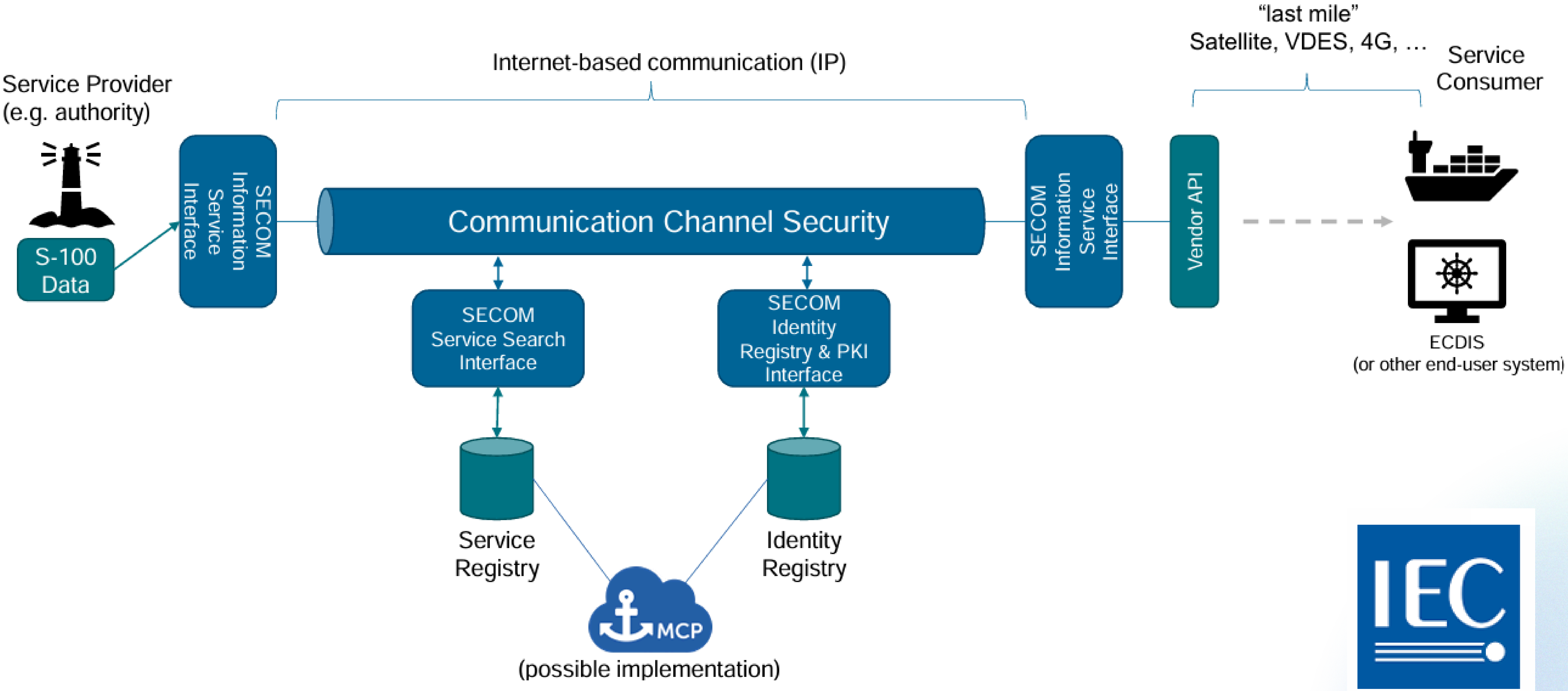
HIERARCHY OBTAINED FROM: ITU-R M2092-1 AND IALA G1139

VDES PURPOSE:

Enhancing Maritime Data Exchange & Safety


1. INCREASE DATA EXCHANGE CAPACITY
2. SUPPORT e-NAVIGATION
3. ENSURE SAFETY PRIORITY
TRADITIONAL AIS (Bottleneck)
MORE DATA
VDES (High Capacity)
DIGITAL SERVICES
AUTOMATION
Digitalization & Process Automation
SAFETY FIRST
EMERGENCY COMMUNICATION
Robust, Structured Emergency Management

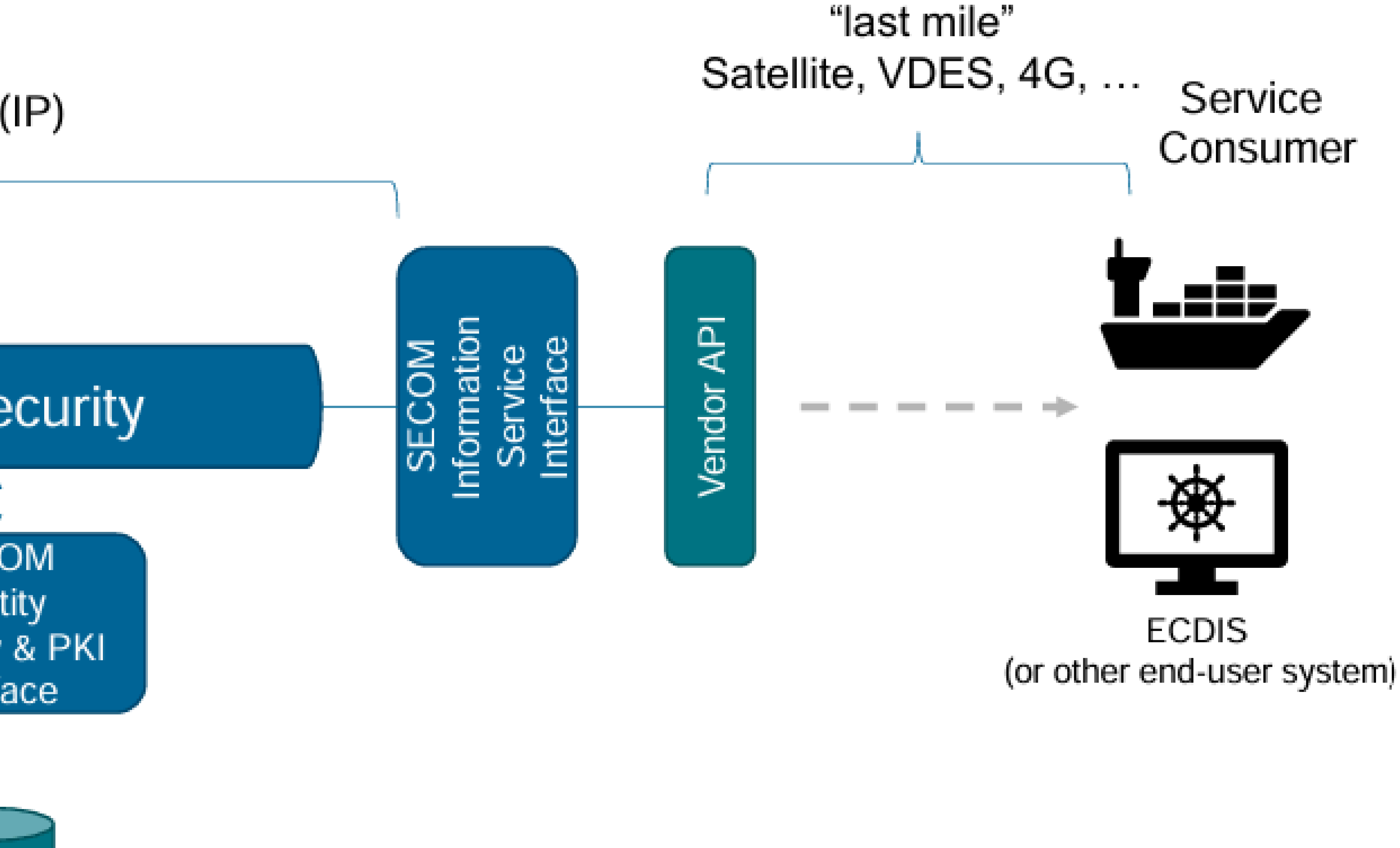
WHAT IS SECOM?



Service Provider
(e.g. authority)
S-100 Data
SECOM Information Service Interface
Internet-based communication (IP)
Communication Channel Security
SECOM Service Search Interface
SECOM Identity Registry & PKI Interface
Service Registry
Identity Registry
MCP
(possible implementation)
SECOM Information Service Interface
Vendor API
"last mile"
Satellite, VDES, 4G, …
Service Consumer
ECDIS
(or other end-user system)

CREDITS TO THESIS: SECOM BETWEEN SHIP AND SHORE CONTROL CENTER USING A MICROSERVICE ARCHITECTURE

VDES AND SECOM INTEGRATION



WHO STANDARDIZES THEM?

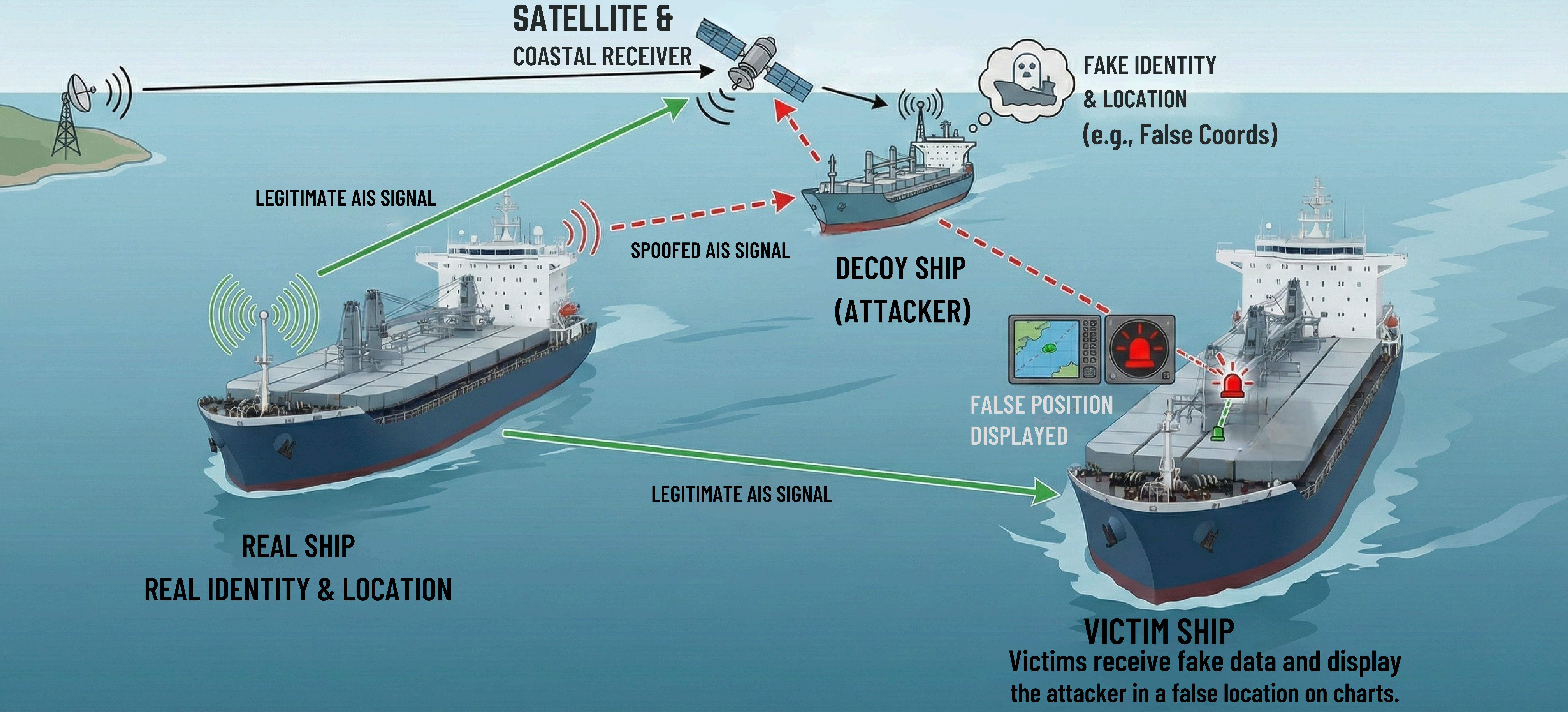
INTERNATIONAL TELECOMMUNICATION UNION

INTERNATIONAL MARITIME ORGANIZATION

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INTERNATIONAL ASSOCIATION OF MARINE AIDS TO NAVIGATION AND LIGHTHOUSE AUTHORITIES

MARITIME SPOOFING ATTACK



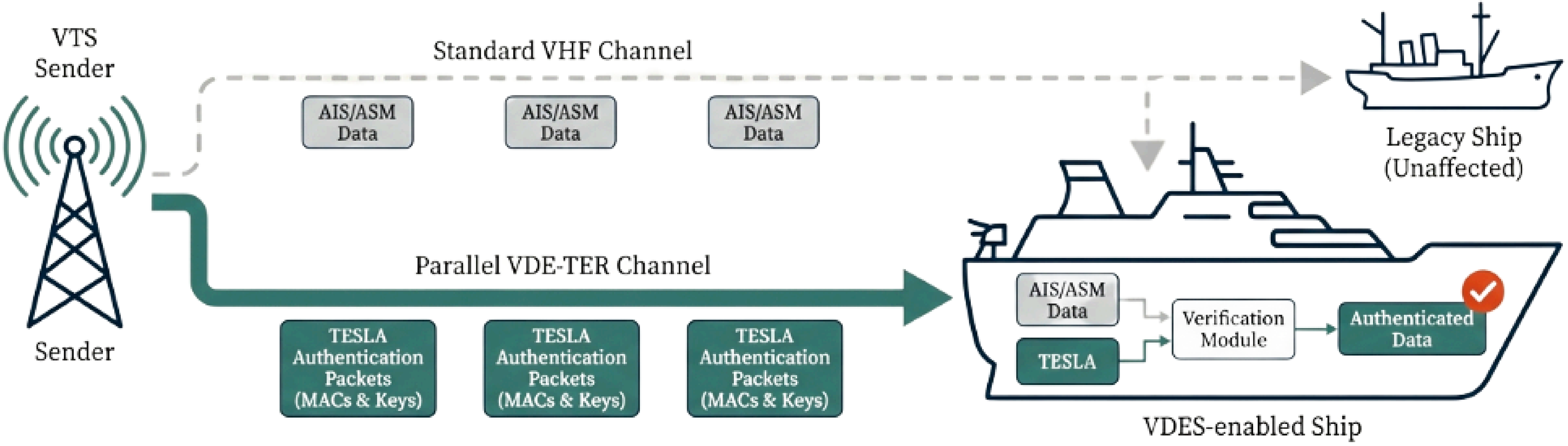
Victims receive fake data and display the attacker in a false location on charts.

SUGGESTED SOLUTION

BY IALA GUIDELINE ON "VDES AUTHENTICATION TECHNIQUES"

Use the **VDE-TER** channel to transmit authentication data in parallel through the TESLA protocol. In this way:

- AIS/ASMs remain unchanged → full compatibility with legacy systems.
- Authentication can travels on a separate channel, without introducing overhead into the original messages.



ADAPTATION OF THE TESLA PROTOCOL

TESLA PROTOCOL PAPER: RFC 4082/2005

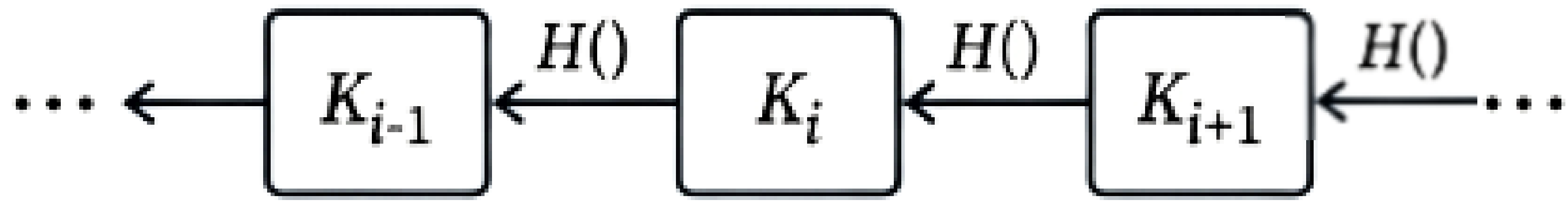
01 Key Chain Generation & Disclosure

02 Bootstrap & Synchronization

03 How the protocol should be integrated

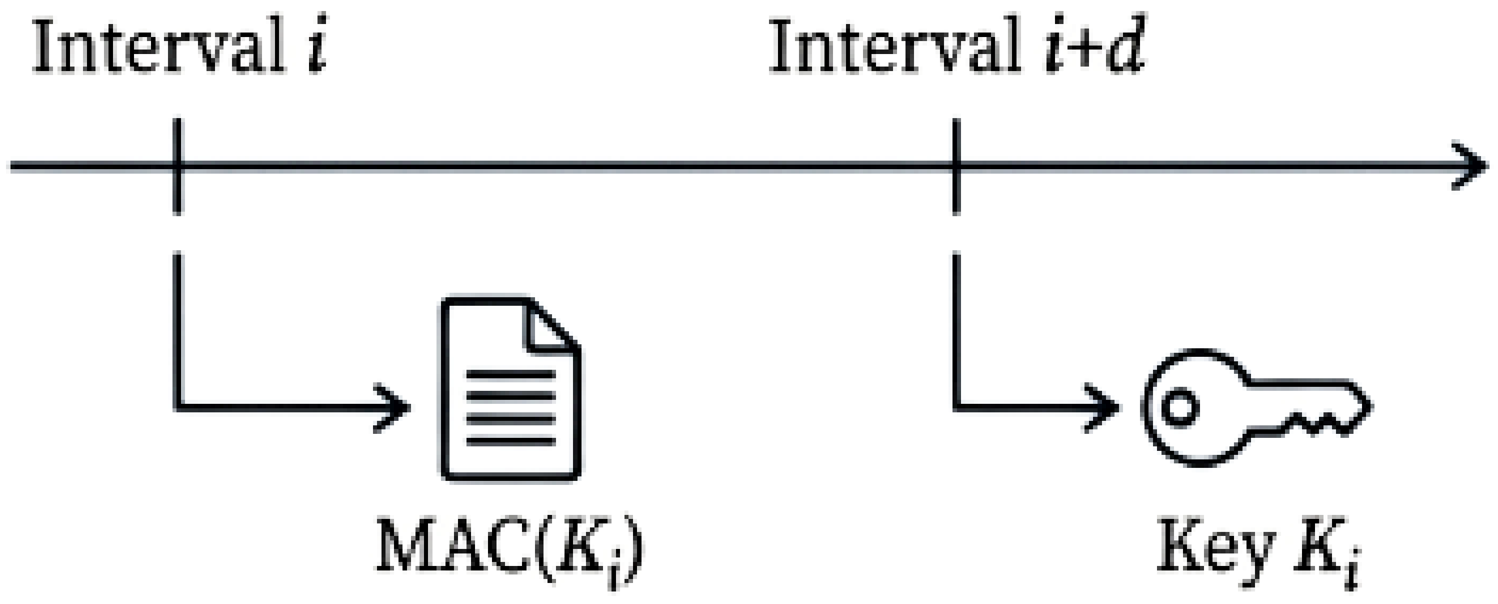
KEY CHAIN GENERATION & DISCLOSURE

TESLA Core Mechanism



$\cdots \leftarrow K_{i-1} \xleftarrow{H()} K_i \xleftarrow{H()} K_{i+1} \xleftarrow{H()} \cdots$

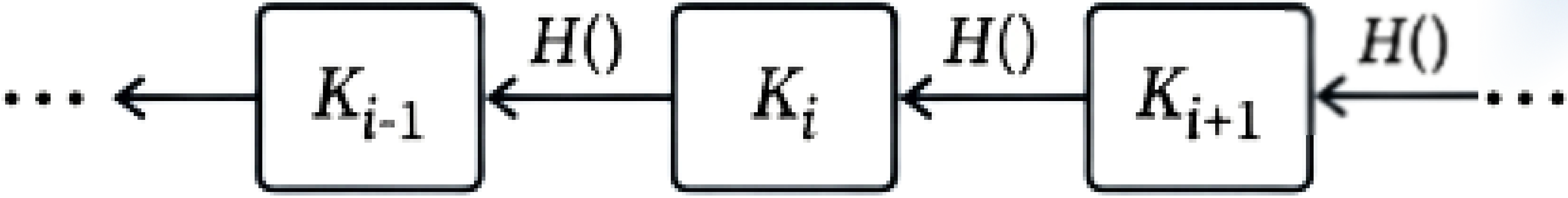
One-Way Key Chain: Keys are generated backward. Easy to verify, impossible to predict.



Delayed Key Disclosure: The key to verify a message is revealed later, preventing real-time forgery.

KEY CHAIN GENERATION & DISCLOSURE

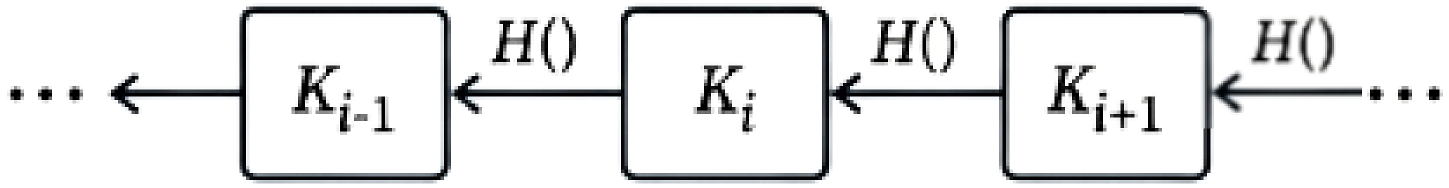
TESLA Core Mechanism



One-Way Key Chain: Keys are generated backward. Easy to verify, impossible to predict.

KEY CHAIN GENERATION & DISCLOSURE

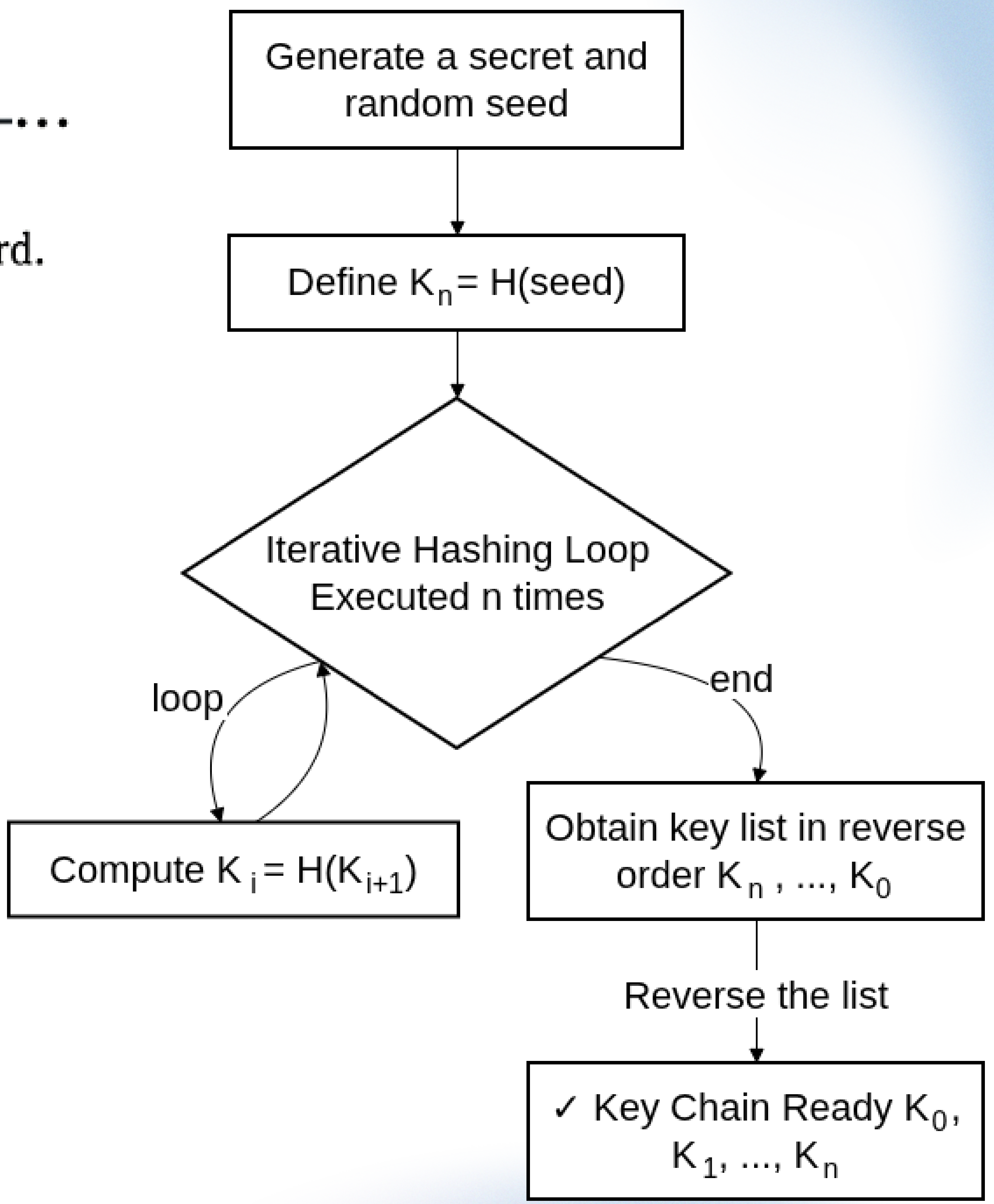
TESLA Core Mechanism



One-Way Key Chain: Keys are generated backward.
Easy to verify, impossible to predict.

USE OF HASH FUNCTION SHA-256

- DETERMINISTIC
- COLLISION RESISTANT
- ONE-WAY



Generate a secret and random seed

Define K_n= H(seed)

Iterative Hashing Loop Executed n times

loop

Compute K $_i$= H(K_{i+1})

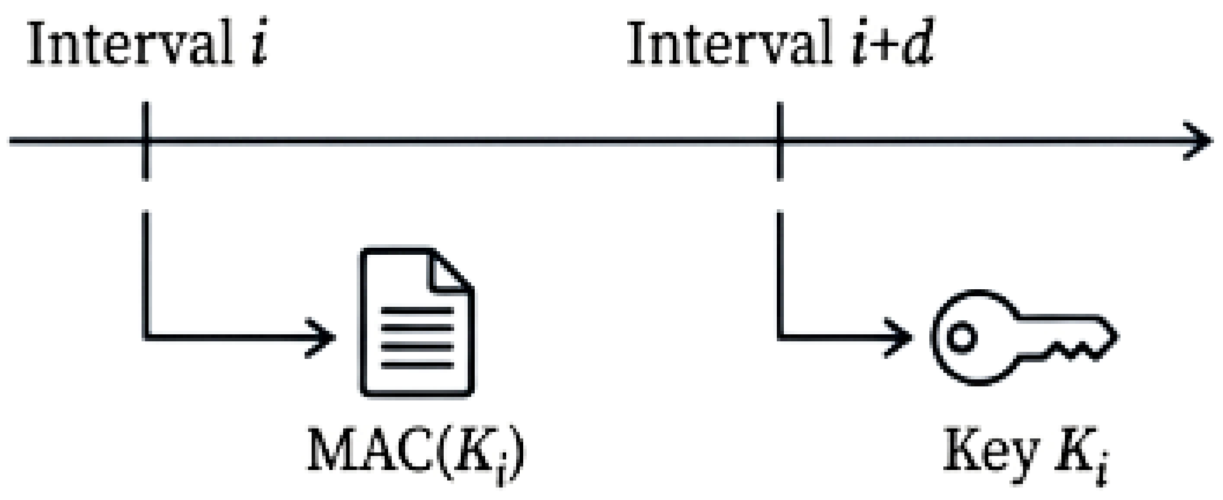
end

Obtain key list in reverse order K_n , ..., K_0

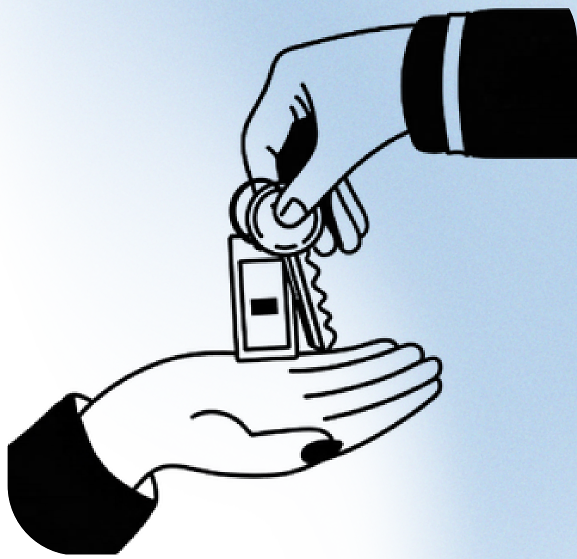
Reverse the list

✓ Key Chain Ready K_0, K_1, ..., K_n

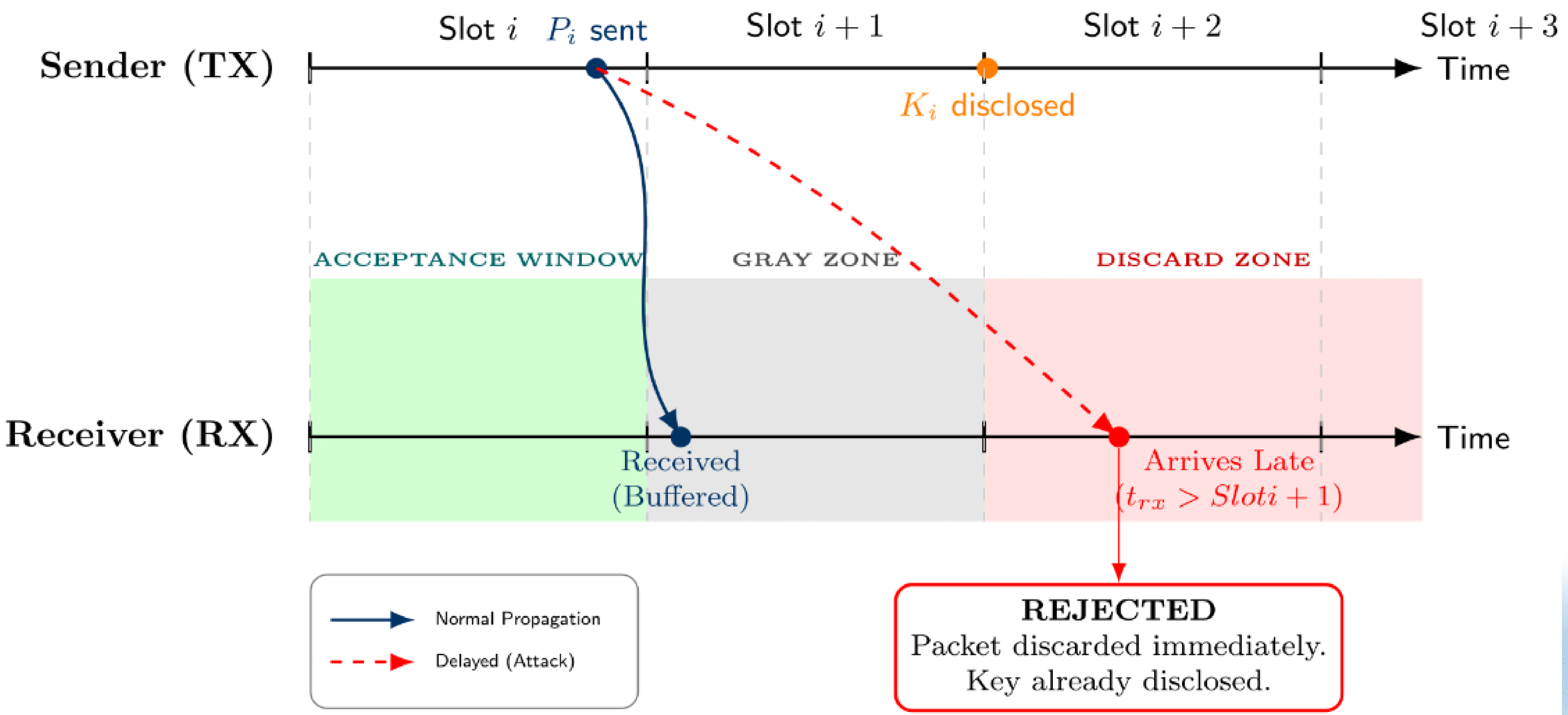
KEY CHAIN GENERATION & DISCLOSURE



Delayed Key Disclosure: The key to verify a message is revealed later, preventing real-time forgery.



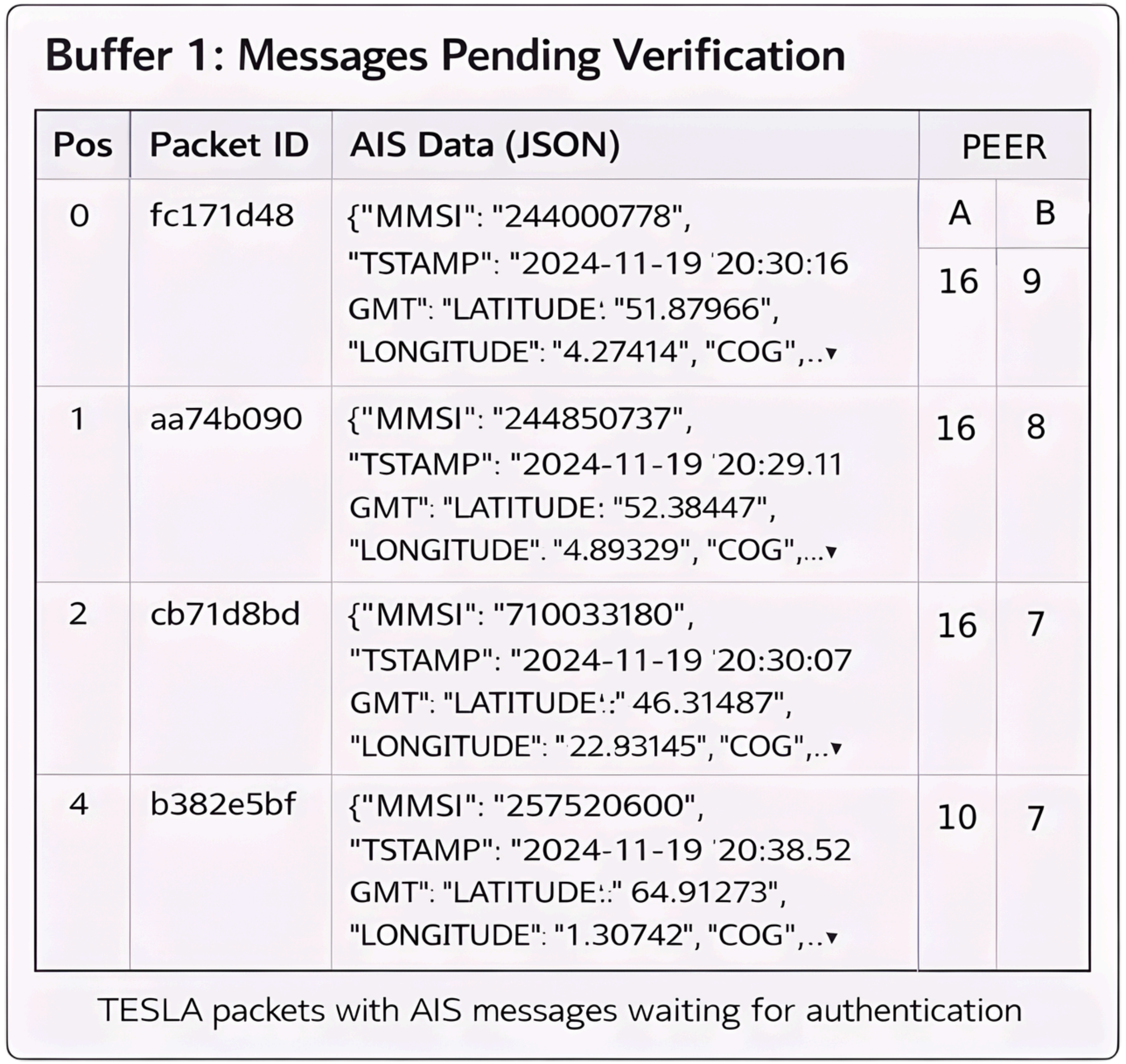
ORDER AND TIME OF RECEIPT ARE FUNDAMENTAL



KEY CHAIN GENERATION & DISCLOSURE

DOUBLE BUFFER LOGIC

AIS CHANNEL FLOW

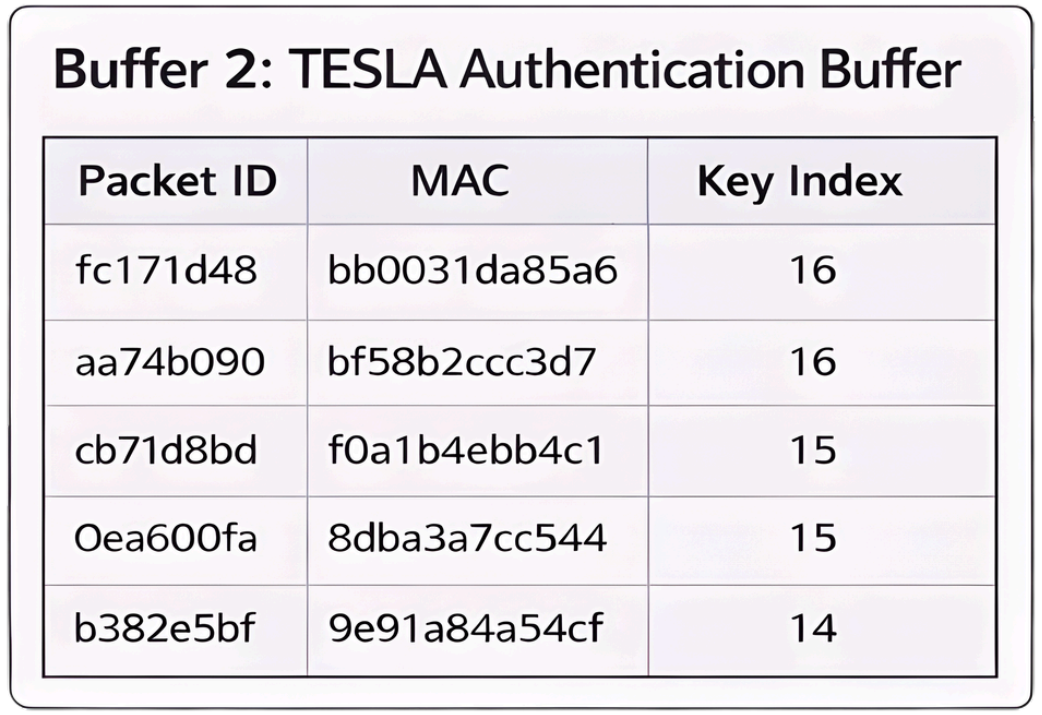


Buffer 1: Messages Pending Verification

Pos	Packet ID	AIS Data (JSON)	PEER	
			A	B
0	fc171d48	{"MMSI": "244000778", "TSTAMP": "2024-11-19 '20:30:16 GMT": "LATITUDE: "51.87966", "LONGITUDE": "4.27414", "COG",..▾	16	9
1	aa74b090	{"MMSI": "244850737", "TSTAMP": "2024-11-19 '20:29.11 GMT": "LATITUDE: "52.38447", "LONGITUDE". "4.89329", "COG",...▾	16	8
2	cb71d8bd	{"MMSI": "710033180", "TSTAMP": "2024-11-19 '20:30:07 GMT": "LATITUDE::" 46.31487", "LONGITUDE": "22.83145", "COG",..▾	16	7
4	b382e5bf	{"MMSI": "257520600", "TSTAMP": "2024-11-19 '20:38.52 GMT": "LATITUDE::" 64.91273", "LONGITUDE": "1.30742", "COG",..▾	10	7

TESLA packets with AIS messages waiting for authentication

VDE-TER CAHNNEL FLOW



Buffer 2: TESLA Authentication Buffer

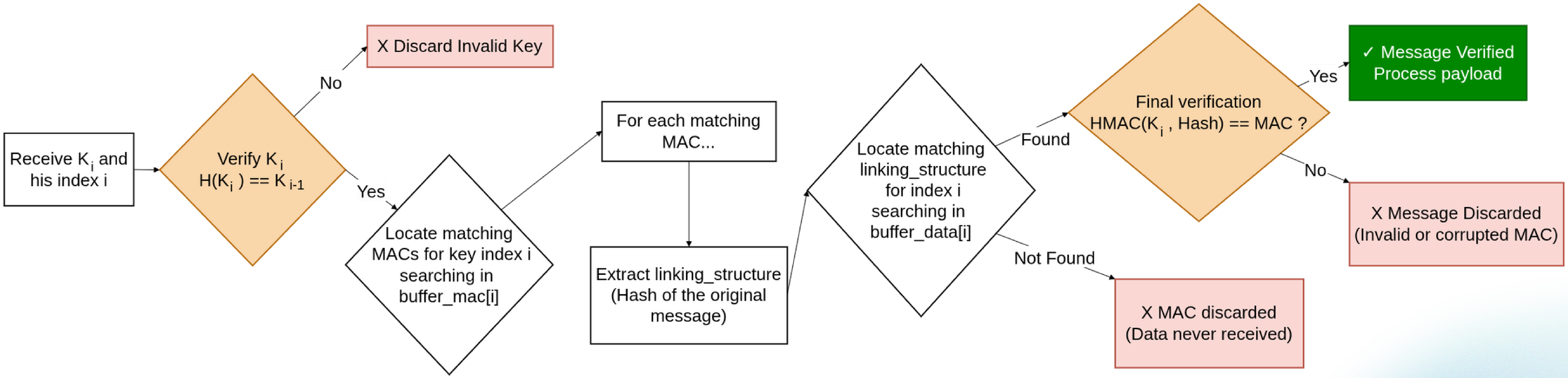
Packet ID	MAC	Key Index
fc171d48	bb0031da85a6	16
aa74b090	bf58b2ccc3d7	16
cb71d8bd	f0a1b4ebb4c1	15
0ea600fa	8dba3a7cc544	15
b382e5bf	9e91a84a54cf	14

MESSAGE VERIFICATION PROCESS

☑ AIS MESSAGE

☑ MAC PACKET

☑ TESLA KEY



Receive K_i and his index i

Verify K_i $H(K_i) == K_{i-1}$

No

X Discard Invalid Key

Yes

Locate matching MACs for key index i searching in buffer_mac[i]

For each matching MAC...

Extract linking_structure (Hash of the original message)

Locate matching linking_structure for index i searching in buffer_data[i]

Found

Not Found

X MAC discarded (Data never received)

Final verification HMAC(K_i , Hash) == MAC ?

Yes

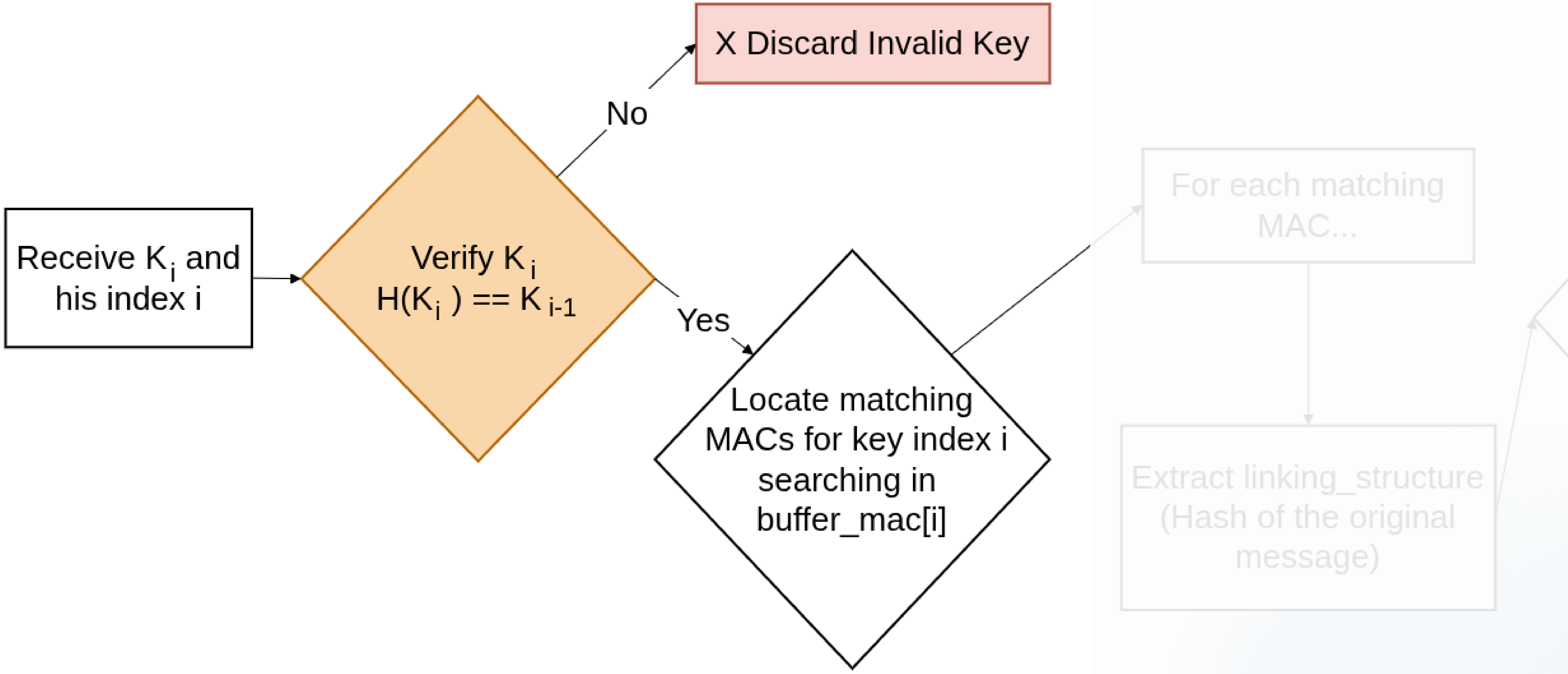
✓ Message Verified Process payload

No

X Message Discarded (Invalid or corrupted MAC)

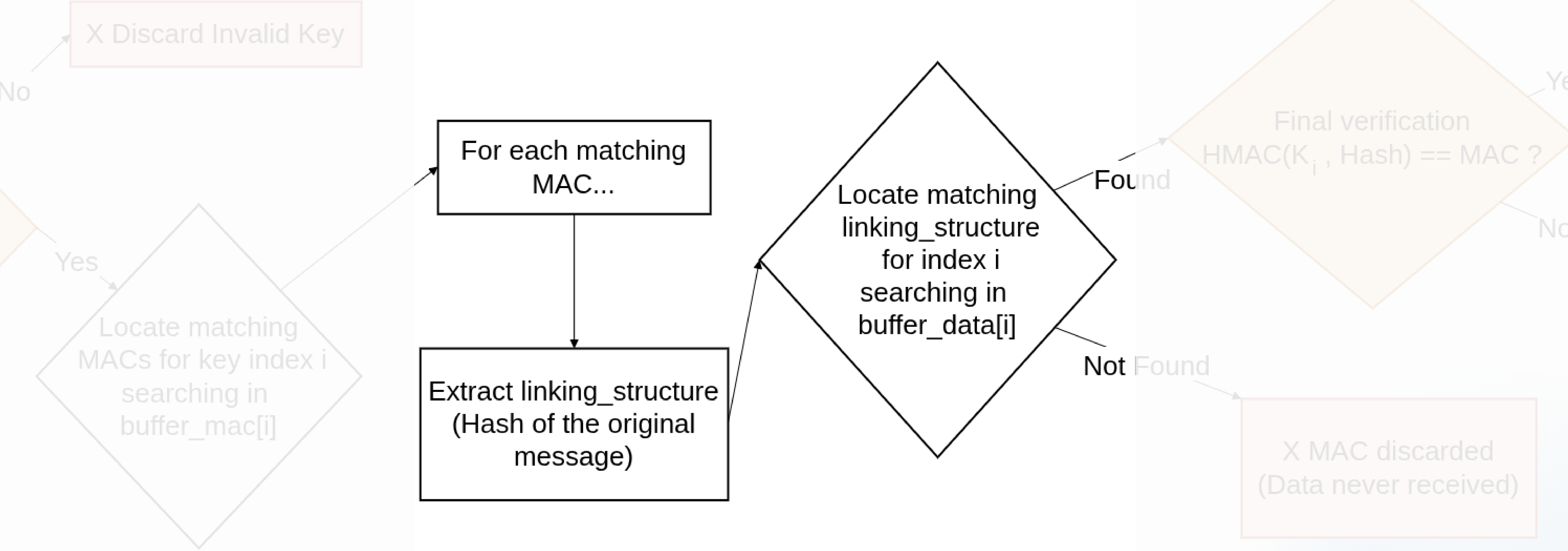
MESSAGE VERIFICATION PROCESS

☑ AIS MESSAGE ☑ MAC PACKET ☑ TESLA KEY


X Discard Invalid Key
No
Receive K_i and his index i
Verify K_i
$H(K_i) == K_{i-1}$
Yes
Locate matching MACs for key index i searching in buffer_mac[i]
For each matching MAC...
Extract linking_structure (Hash of the original message)

MESSAGE VERIFICATION PROCESS

☑ AIS MESSAGE ☑ MAC PACKET ☑ TESLA KEY



X Discard Invalid Key

No

Yes

Locate matching MACs for key index i searching in buffer_mac[i]

For each matching MAC...

Extract linking_structure (Hash of the original message)

Locate matching linking_structure for index i searching in buffer_data[i]

Found

Not Found

Final verification HMAC(K_i , Hash) == MAC ?

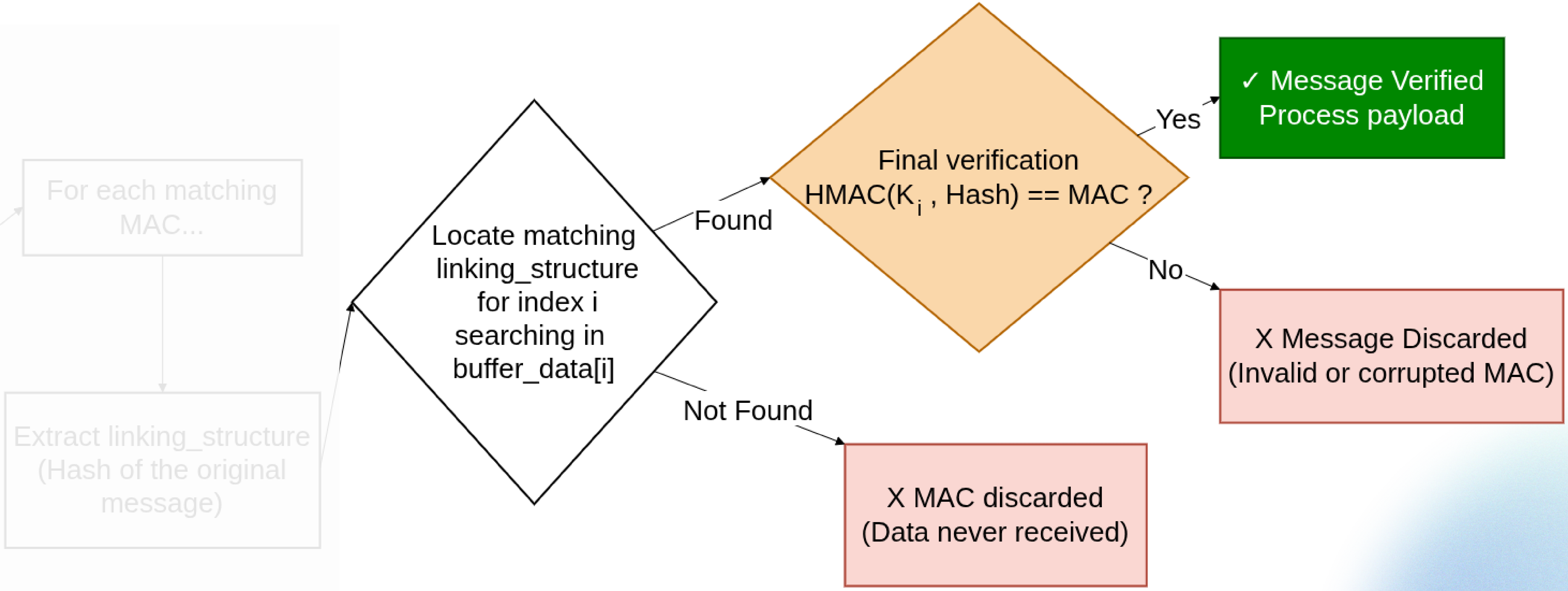
Ye

No

X MAC discarded (Data never received)

MESSAGE VERIFICATION PROCESS

☑ AIS MESSAGE ☑ MAC PACKET ☑ TESLA KEY



For each matching MAC...

Extract linking_structure (Hash of the original message)

Locate matching linking_structure for index i searching in buffer_data[i]

Found

Not Found

Final verification HMAC(K_i , Hash) == MAC ?

Yes

No

✓ Message Verified Process payload

X Message Discarded (Invalid or corrupted MAC)

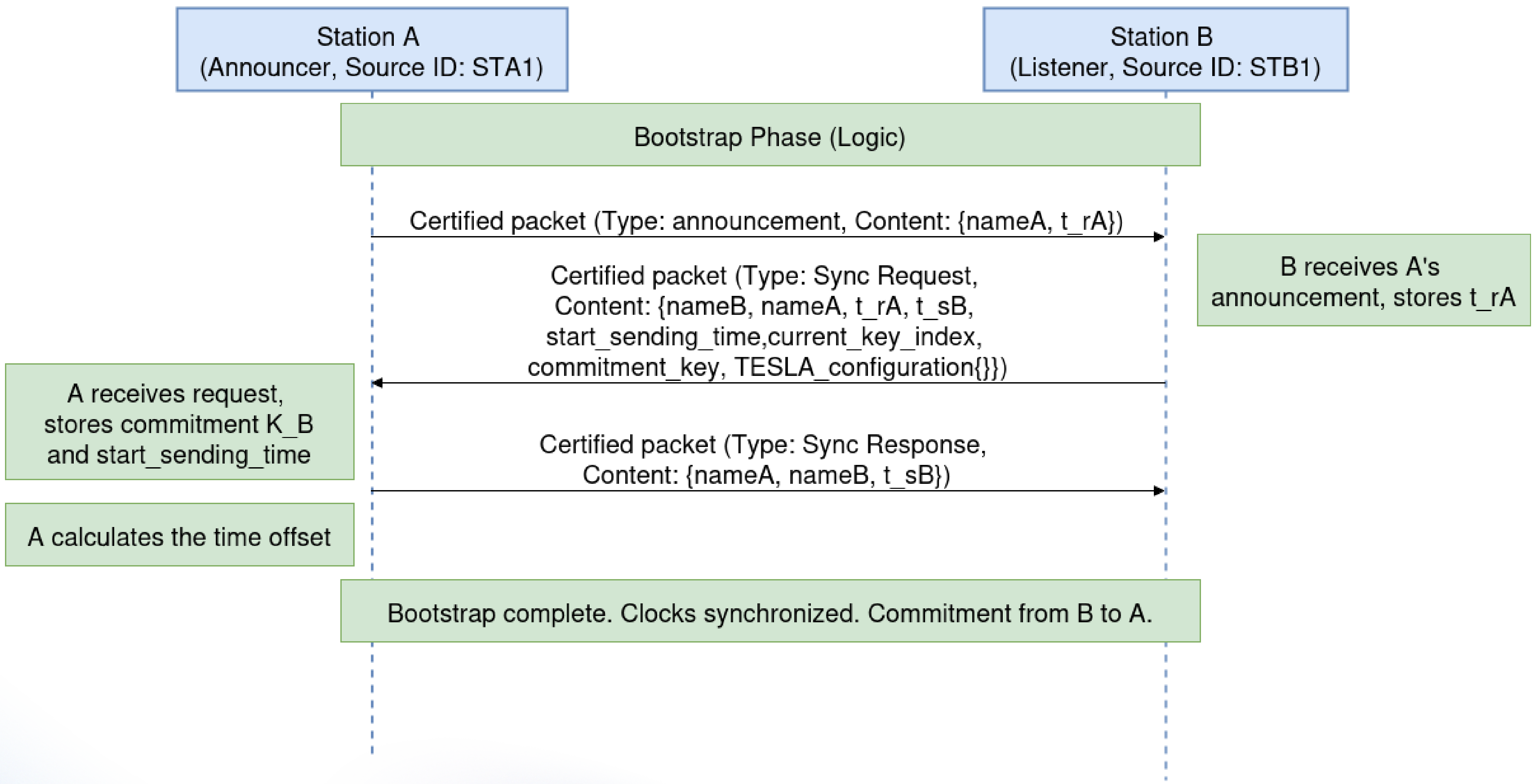
X MAC discarded (Data never received)

MESSAGE VERIFICATION PROCESS

HOW DO WE SECURELY SHARE FIRST KEY AND SYNCHRONIZE CLOCKS?

HOW DO WE SECURELY SHARE FIRST KEY AND SYNCHRONIZE CLOCKS?

BOOTSTRAP PHASE = THREE WAY HANDSHAKE SECURED WITH PKI



Station A
(Announcer, Source ID: STA1)

Station B
(Listener, Source ID: STB1)

Bootstrap Phase (Logic)

Certified packet (Type: announcement, Content: {nameA, t_rA})

B receives A's announcement, stores t_rA

Certified packet (Type: Sync Request, Content: {nameB, nameA, t_rA, t_sB, start_sending_time,current_key_index, commitment_key, TESLA_configuration{}})

A receives request, stores commitment K_B and start_sending_time

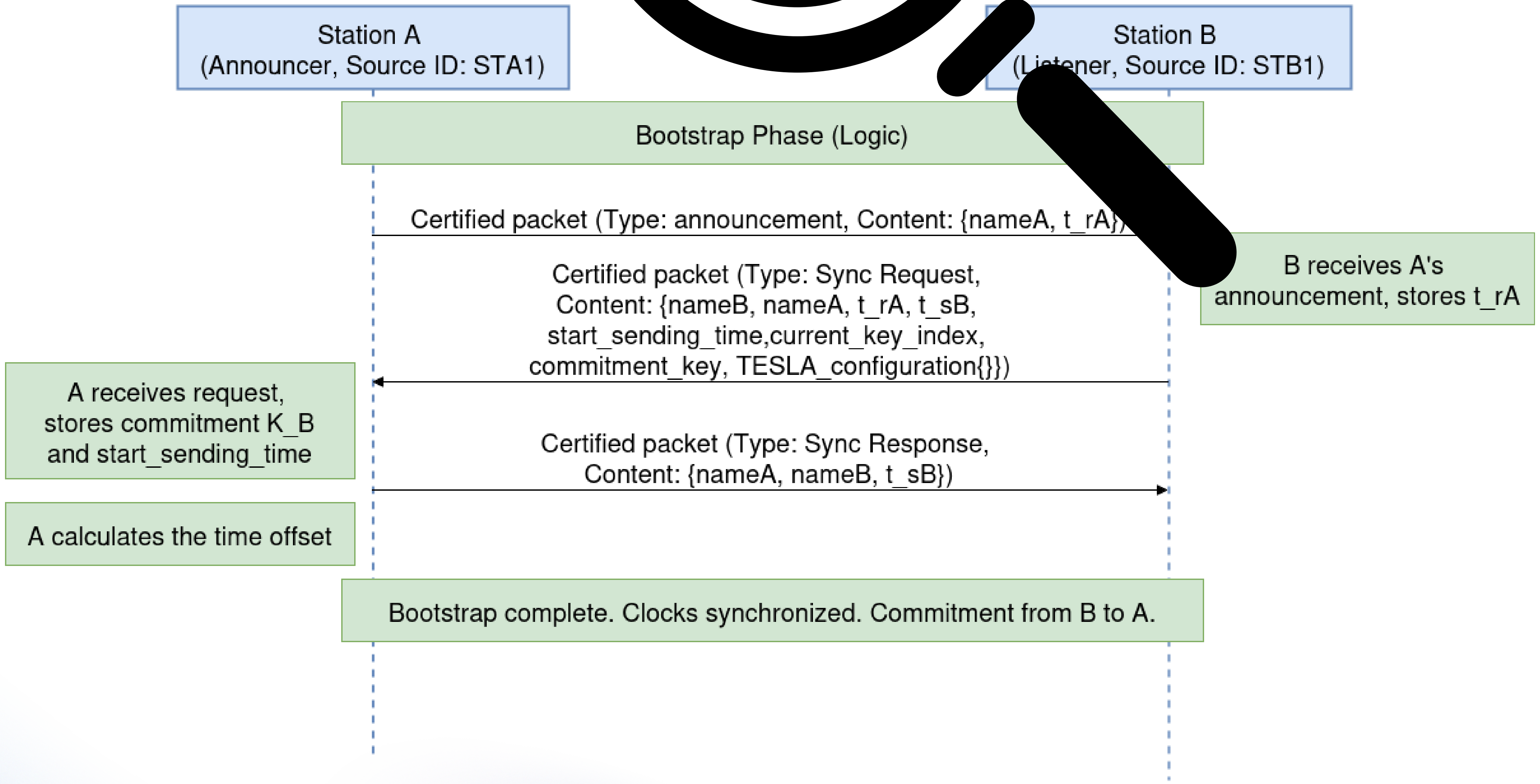
Certified packet (Type: Sync Response, Content: {nameA, nameB, t_sB})

A calculates the time offset

Bootstrap complete. Clocks synchronized. Commitment from B to A.

HOW DO WE SECURELY SHARE FIRST KEY AND SYNCHRONIZE CLOCKS?

BOOTSTRAP PHASE = THREE WAY HANDSHAKE SECURED WITH PKI



Station A
(Announcer, Source ID: STA1)

Station B
(Listener, Source ID: STB1)

Bootstrap Phase (Logic)

Certified packet (Type: announcement, Content: {nameA, t_rA})

B receives A's announcement, stores t_rA

Certified packet (Type: Sync Request, Content: {nameB, nameA, t_rA, t_sB, start_sending_time,current_key_index, commitment_key, TESLA_configuration{}})

A receives request, stores commitment K_B and start_sending_time

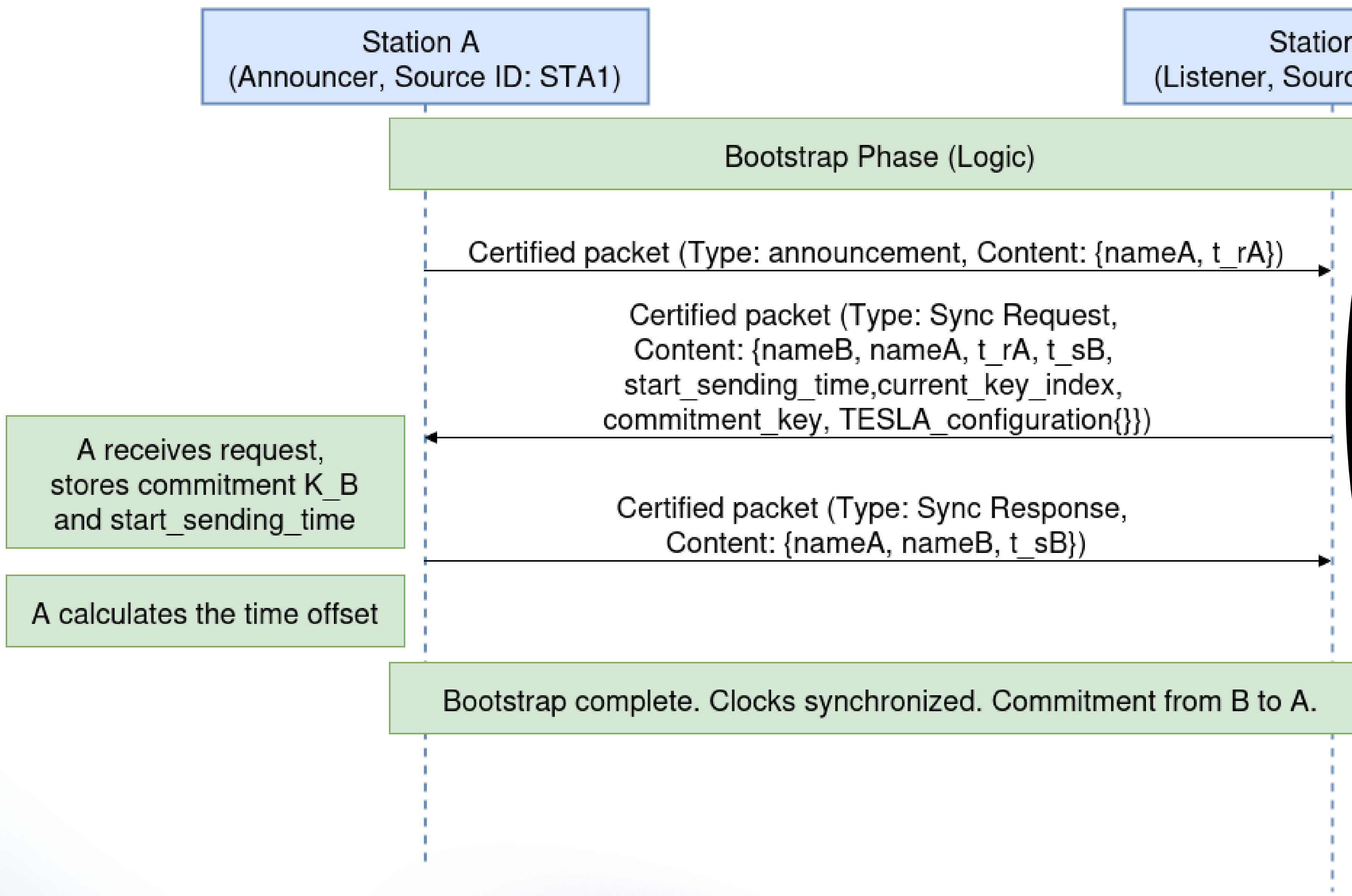
Certified packet (Type: Sync Response, Content: {nameA, nameB, t_sB})

A calculates the time offset

Bootstrap complete. Clocks synchronized. Commitment from B to A.

HOW DO WE SECURELY SHARE FIRST KEY AND SYNCHRONIZE CLOCKS?

BOOTSTRAP PHASE = THREE WAY HANDSHAKE SECURED WITH PKI



Station A
(Announcer, Source ID: STA1)

Station B
(Listener, Source ID

Bootstrap Phase (Logic)

Certified packet (Type: announcement, Content: {nameA, t_rA})

receives
cement, s t_rA

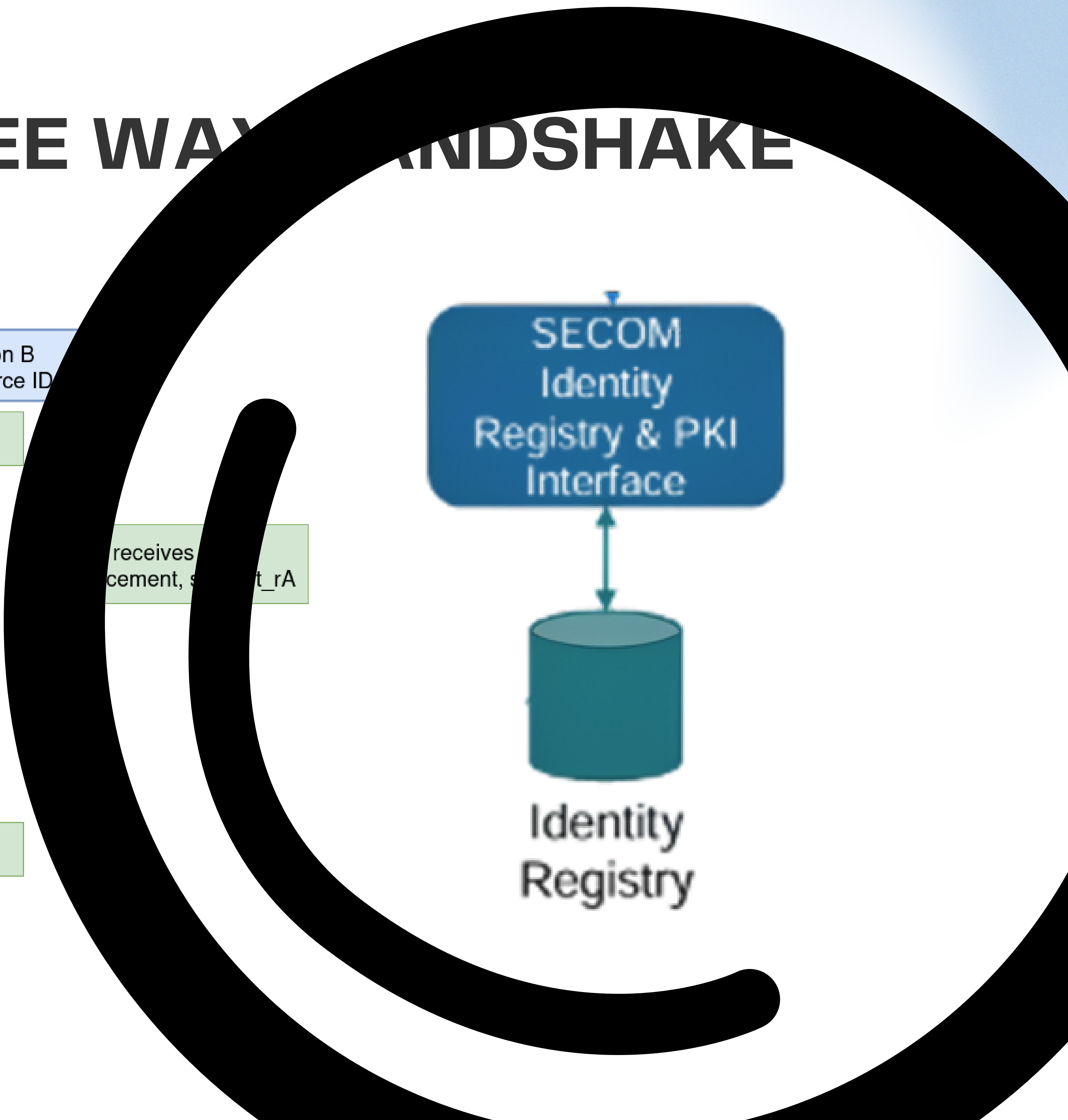
Certified packet (Type: Sync Request,
Content: {nameB, nameA, t_rA, t_sB,
start_sending_time,current_key_index,
commitment_key, TESLA_configuration{}})

A receives request,
stores commitment K_B
and start_sending_time

Certified packet (Type: Sync Response,
Content: {nameA, nameB, t_sB})

A calculates the time offset

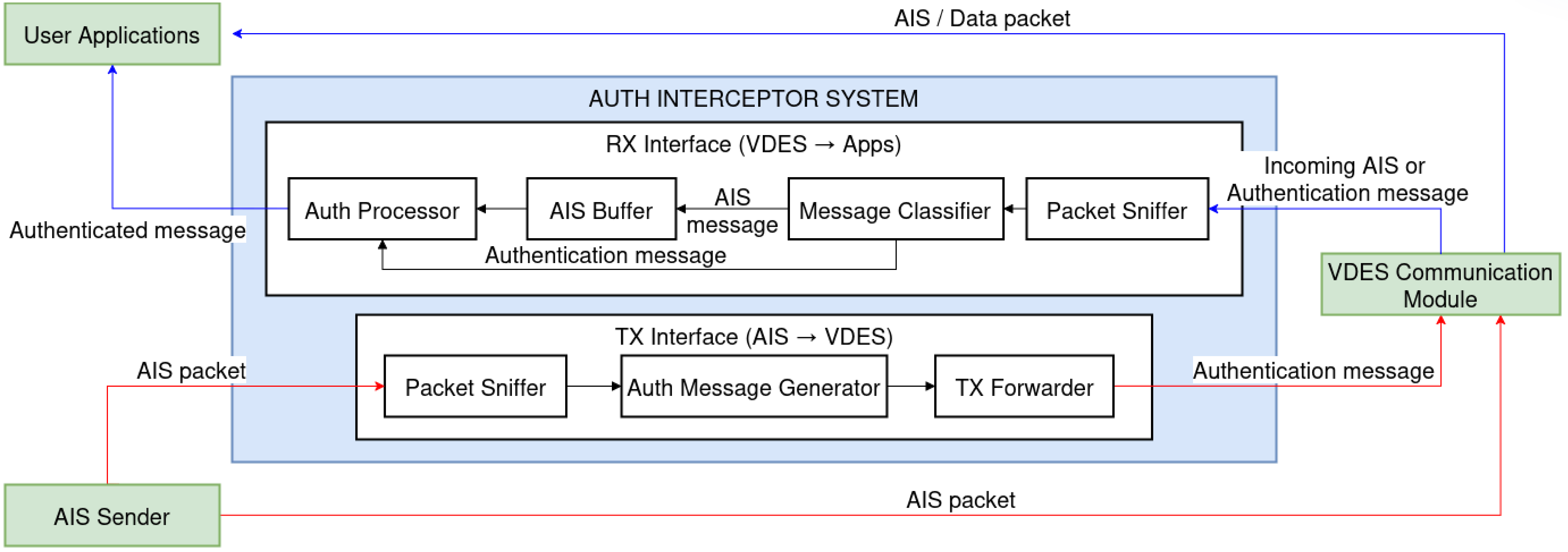
Bootstrap complete. Clocks synchronized. Commitment from B to A.



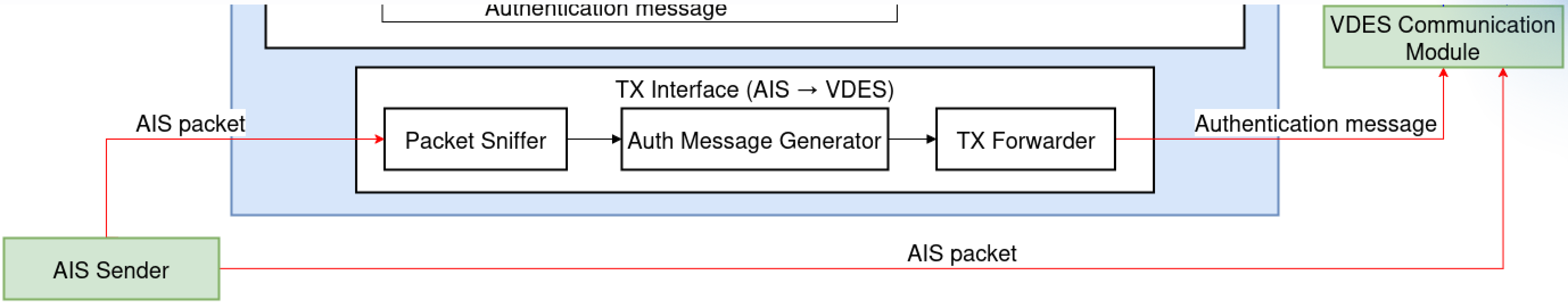
HOW THE PROTOCOL SHOULD BE INTEGRATED?

HOW THE PROTOCOL SHOULD BE INTEGRATED?

THROUGH AN INTERCEPTOR AND AUTHENTICATION SYSTEM



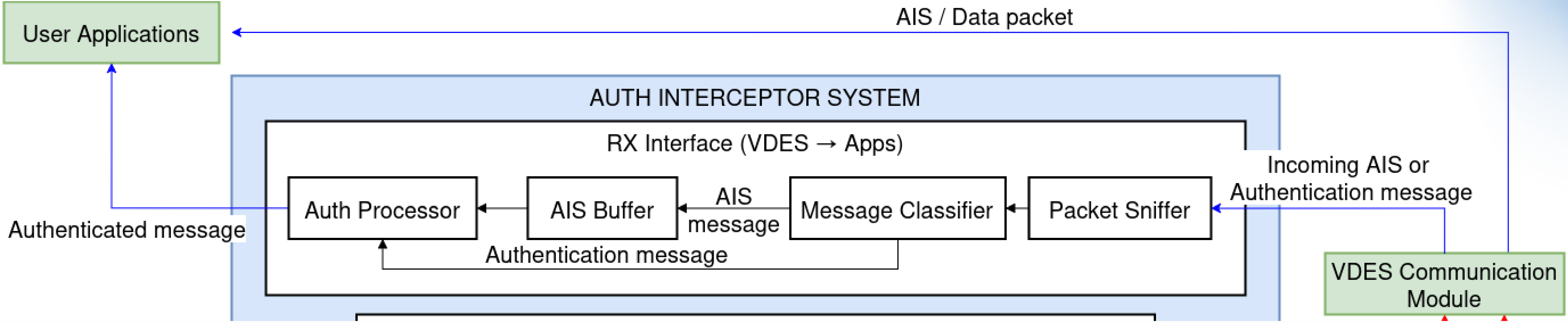
HOW THE PROTOCOL SHOULD BE INTEGRATED?



SENDER INTERFACE(TX) → SENDS PACKETS WITH:

- SNIFF PACKETS COMING FROM AIS (ONBOARD UDP PACKETS)
- SEND AUTHENTICATION MESSAGES (THROUGH VDE-TER)

HOW THE PROTOCOL SHOULD BE INTEGRATED?



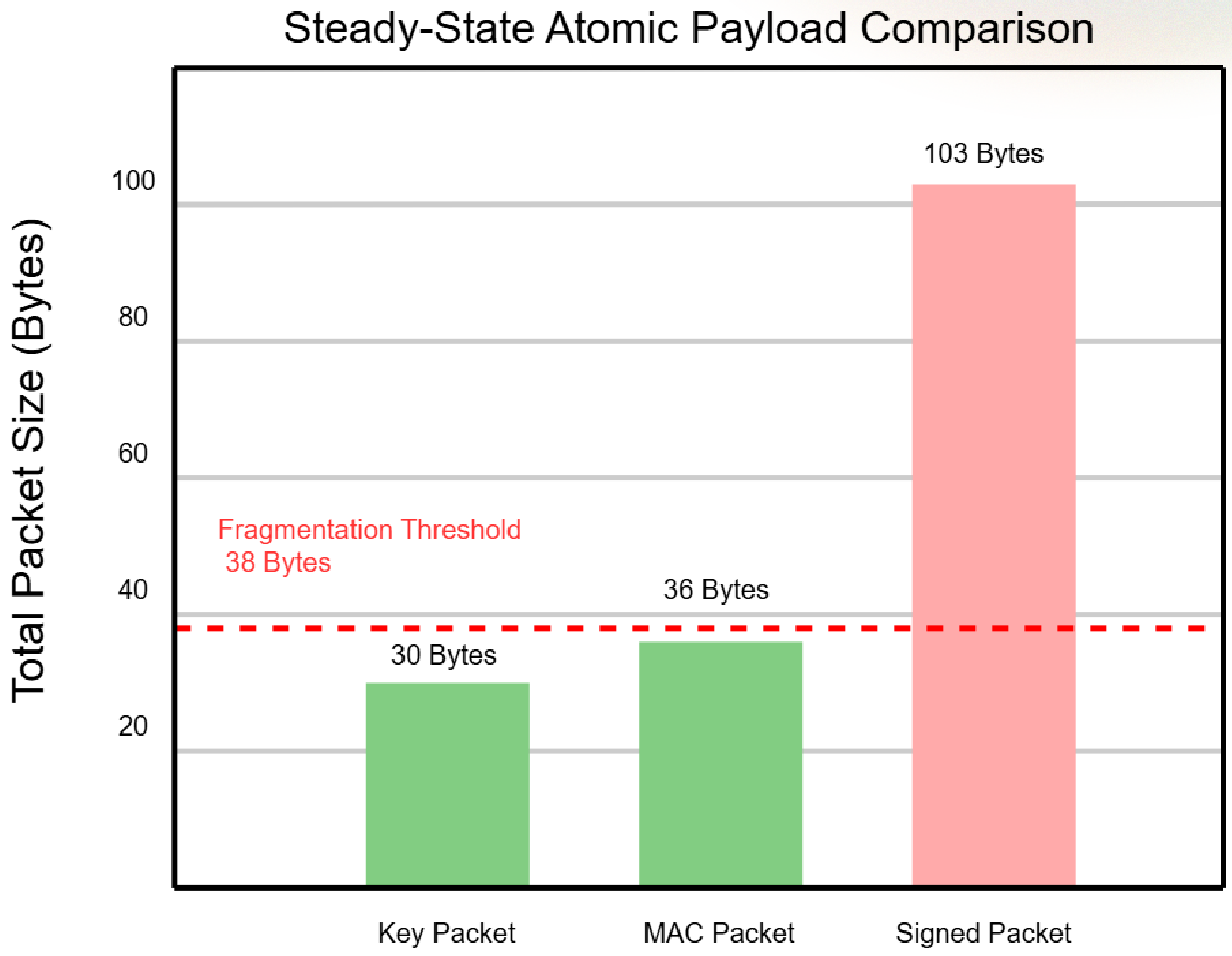
RECEIVER INTERFACE (RX):

- BUFFERS PACKETS WHILE WAITING FOR THE CORRESPONDING KEY
- VERIFY THAT THE OBTAINED KEY BELONGS TO THE CHAIN
- VALIDATES THE MAC
- ACCEPTS THE PACKET AS AUTHENTIC
- SEND THE PACKET AS VALID TO THE USER APPLICATION

PERFORMANCE: TESLA VS. PKI BENCHMARKS

(BASED ON INTEL I5 10TH)

- SCALABILITY WITH 500 VESSELS: <1% CPU VS. >200% FOR PKI.
- DOS RESILIENCE
- PACKET ATOMICITY: 1 PACKET VS. 3 FRAGMENTS REQUIRED BY PKI
- 3-6 SECONDS AUTHENTICATION DELAY, COMPATIBLE WITH AIS REFRESH CYCLES (2-10S)



TESTBED IMPLEMENTATION AND PROTOTYPING

- **AUTH INTERCEPTOR**: PYTHON MIDDLEWARE BETWEEN VDES AND APPLICATIONS
- **IEC 61162-450**: UDP MULTICAST (BINARY IMAGE TRANSFER)
- **HIL LORA (868 MHZ)**: VDES-EQUIVALENT BITRATE, LATENCY AND COLLISIONS
- **DOCKERIZED MICROSERVICES**: PROTOCOL LOGIC & HARDWARE ABSTRACTION
- **AIS SIMULATOR**: SYNTHETIC TRAFFIC FOR LOAD TESTING

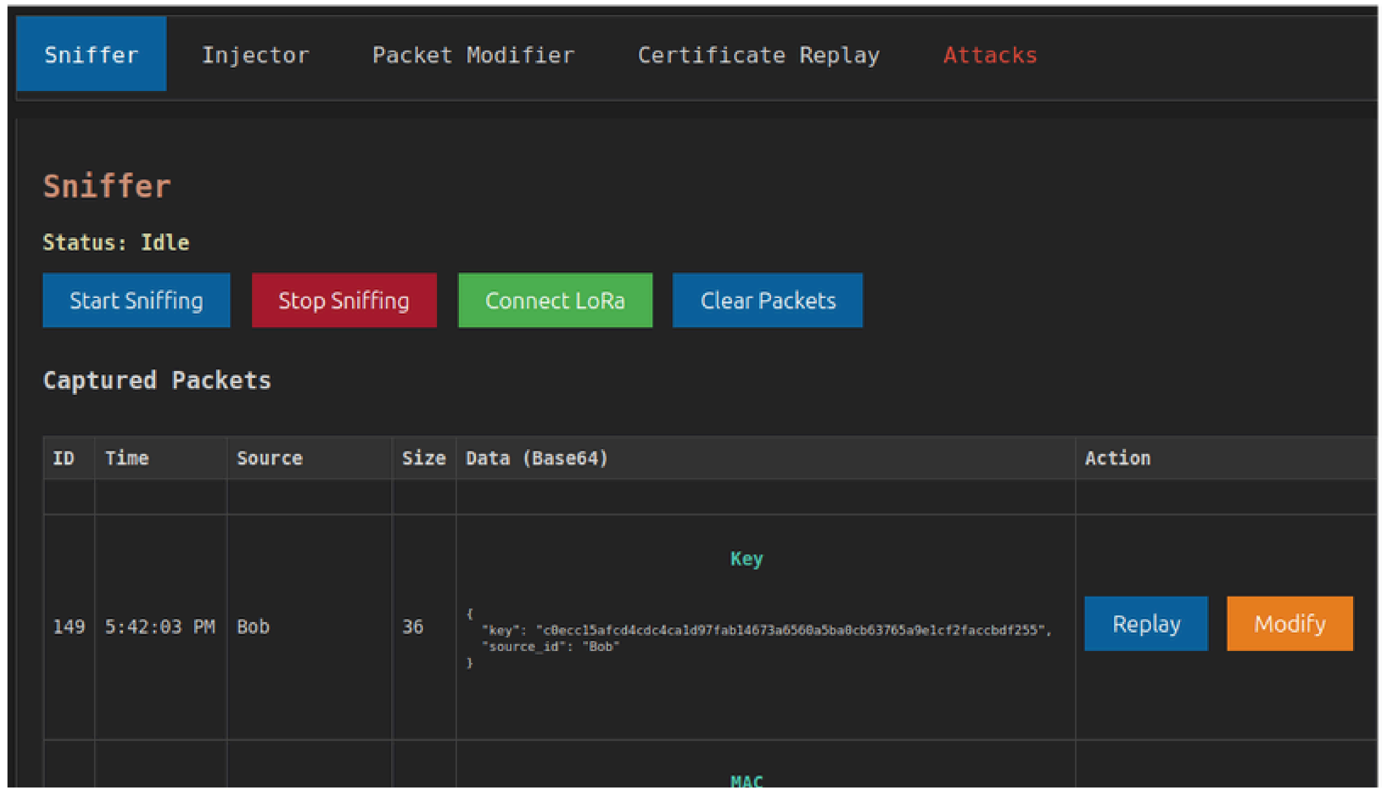
SECURITY VALIDATION AND ATTACK ANALYSIS

RED TEAM FRAMEWORK

- EVIL-VDES: OFFENSIVE MODULE OVER LORA
- SNIFFING & REASSEMBLY

ATTACK VECTORS

- REPLAY & MASQUERADING
- PACKET FORGING
- DOS & JAMMING

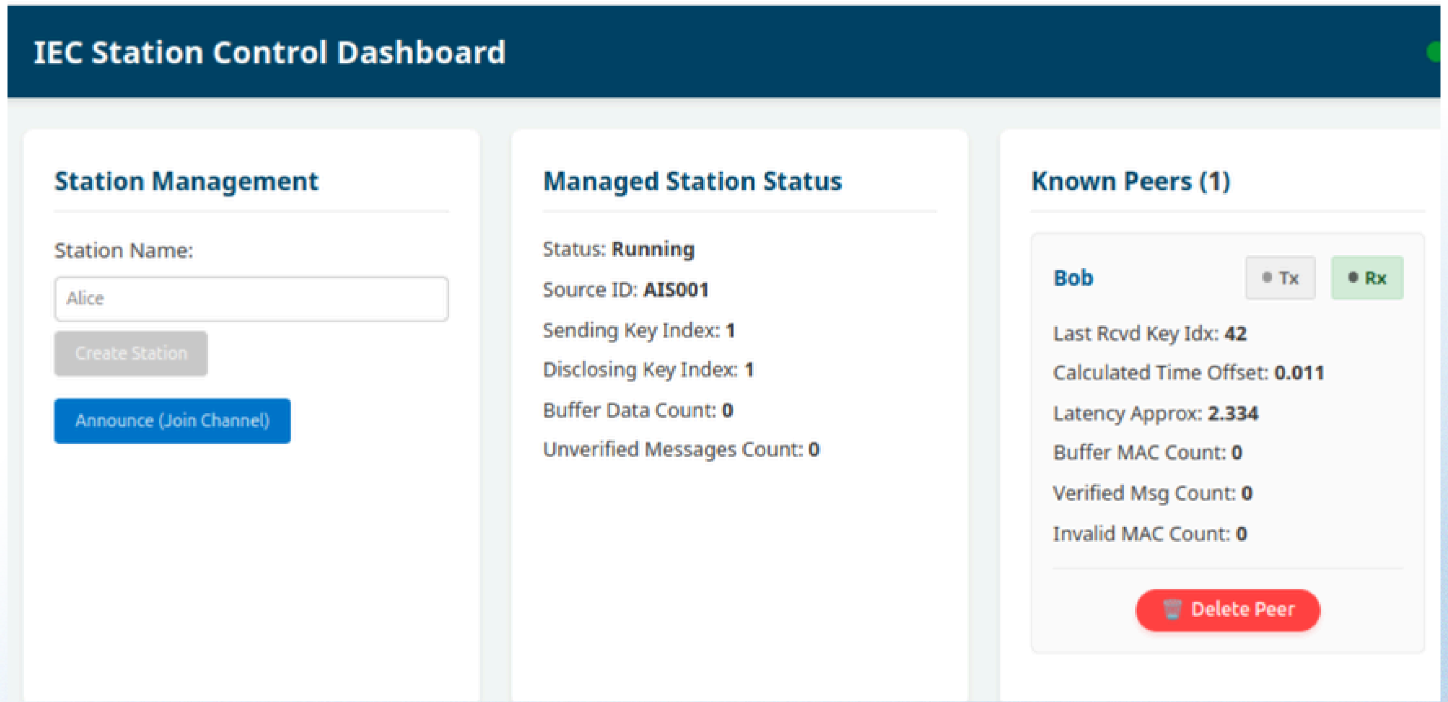


Sniffer

Status: Idle

Captured Packets

ID	Time	Source	Size	Data (Base64)	Action
149	5:42:03 PM	Bob	36	Key { "key": "c0ecc15afcd4cdc4ca1d97fab14673a6560a5ba0cb63765a9e1cf2faccbdf255", "source_id": "Bob" }	Replay Modify
				MAC	



IEC Station Control Dashboard

Station Management

Station Name: Alice

Managed Station Status

Status: **Running**
Source ID: **AIS001**
Sending Key Index: **1**
Disclosing Key Index: **1**
Buffer Data Count: **0**
Unverified Messages Count: **0**

Known Peers (1)

Bob ● Tx ● Rx

Last Rcvd Key Idx: **42**
Calculated Time Offset: **0.011**
Latency Approx: **2.334**
Buffer MAC Count: **0**
Verified Msg Count: **0**
Invalid MAC Count: **0**

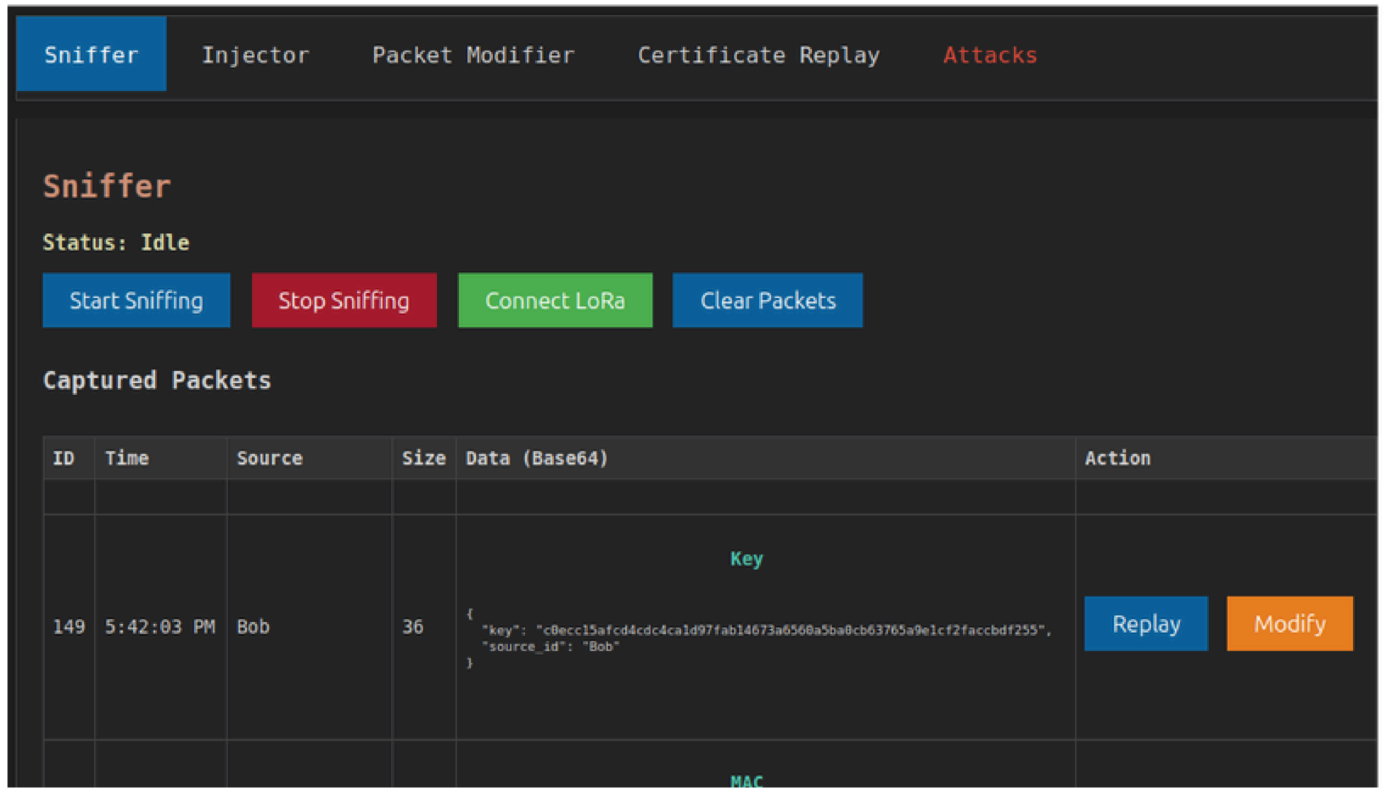
SECURITY VALIDATION AND ATTACK ANALYSIS

RED TEAM FRAMEWORK

- EVIL-VDES: OFFENSIVE MODULE OVER LORA
- SNIFFING & REASSEMBLY

ATTACK VECTORS

- REPLAY & MASQUERADING
- PACKET FORGING
- DOS & JAMMING



Sniffer

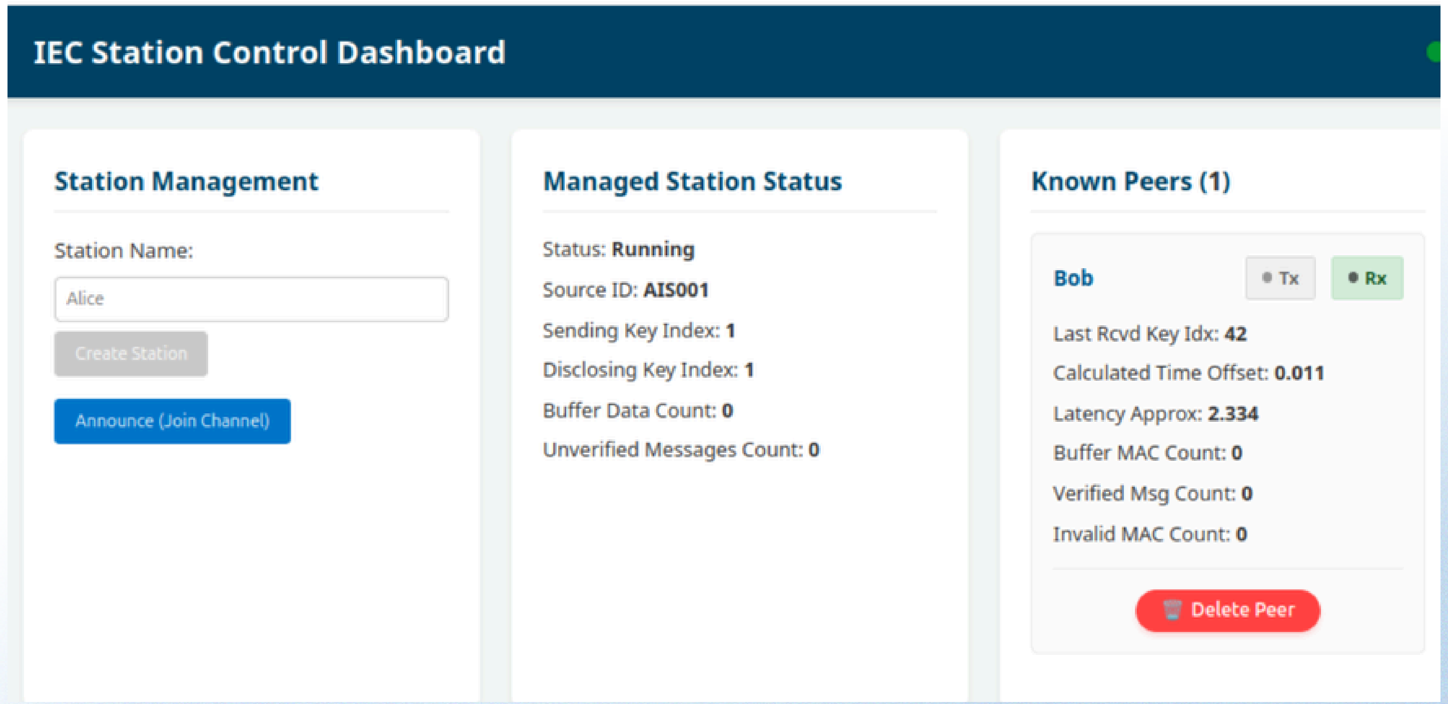
Status: Idle

Captured Packets

ID	Time	Source	Size	Data (Base64)	Action

149	5:42:03 PM	Bob	36	Key
				{
				"key": "c0ecc15afcd4cdc4ca1d97fab14673a6560a5ba0cb63765a9e1cf2faccbdf255",
				"source_id": "Bob"
				}
					Replay Modify

				MAC	


IEC Station Control Dashboard

Station Management

Station Name: Alice

Managed Station Status

Status: **Running**
Source ID: **AIS001**
Sending Key Index: **1**
Disclosing Key Index: **1**
Buffer Data Count: **0**
Unverified Messages Count: **0**

Known Peers (1)

Bob ● Tx ● Rx

Last Rcvd Key Idx: **42**
Calculated Time Offset: **0.011**
Latency Approx: **2.334**
Buffer MAC Count: **0**
Verified Msg Count: **0**
Invalid MAC Count: **0**

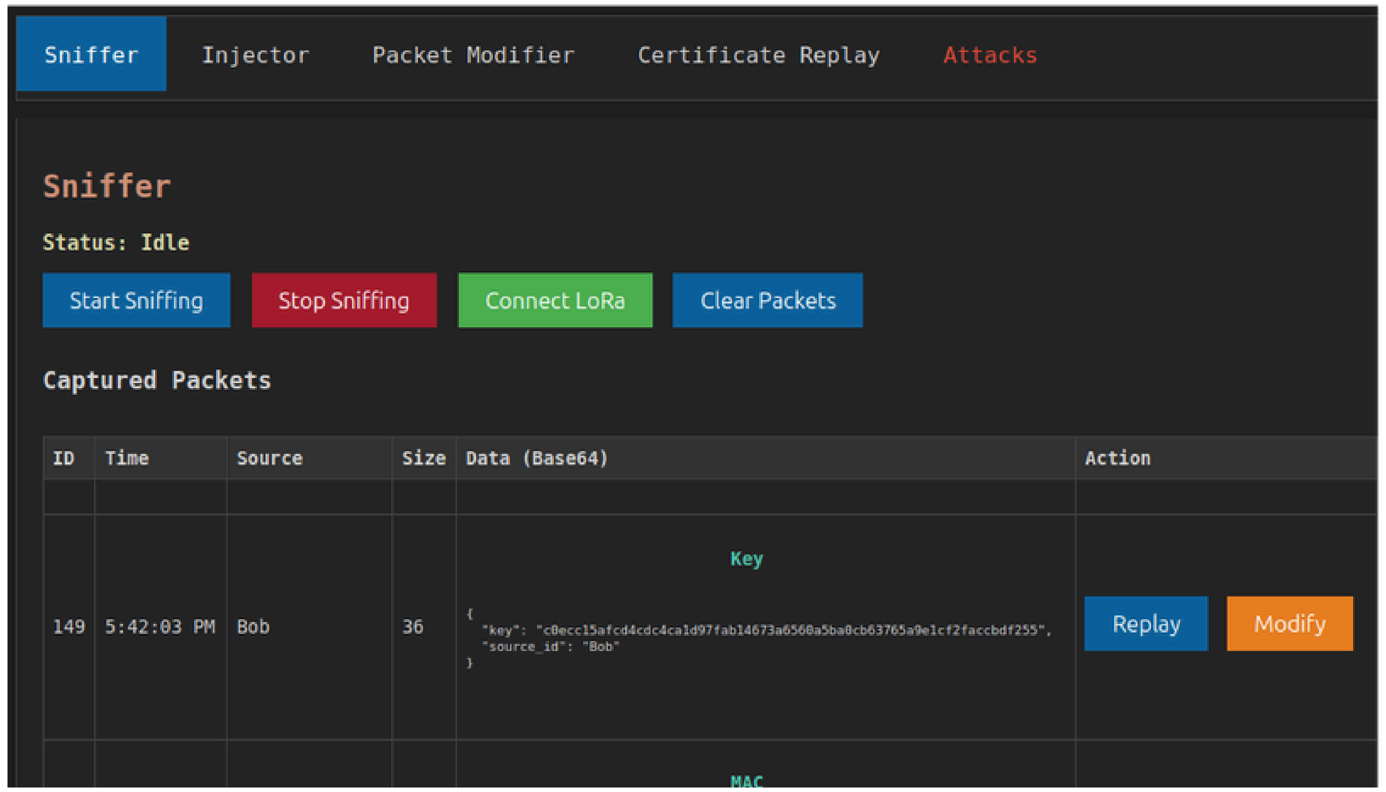
SECURITY VALIDATION AND ATTACK ANALYSIS

RED TEAM FRAMEWORK

- EVIL-VDES: OFFENSIVE MODULE OVER LORA
- SNIFFING & REASSEMBLY

ATTACK VECTORS

- REPLAY & MASQUERADING
- PACKET FORGING
- DOS & JAMMING



Sniffer

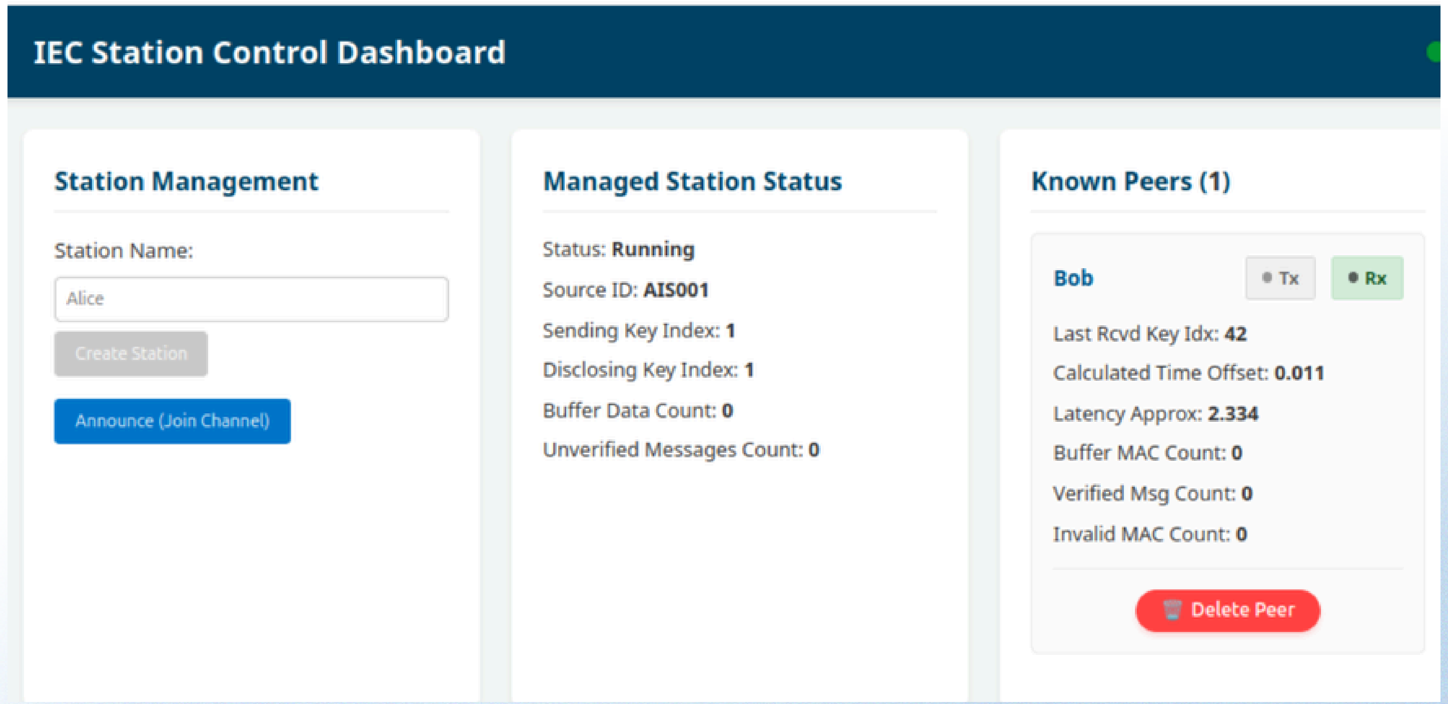
Status: Idle

Captured Packets

ID	Time	Source	Size	Data (Base64)	Action

149	5:42:03 PM	Bob	36	Key
				{
				"key": "c0ecc15afcd4cdc4ca1d97fab14673a6560a5ba0cb63765a9e1cf2faccbdf255",
				"source_id": "Bob"
				}
					Replay Modify

				MAC	



IEC Station Control Dashboard

Station Management

Station Name: Alice

Managed Station Status

Status: Running
Source ID: AIS001
Sending Key Index: 1
Disclosing Key Index: 1
Buffer Data Count: 0
Unverified Messages Count: 0

Known Peers (1)

Bob

Last Rcvd Key Idx: 42
Calculated Time Offset: 0.011
Latency Approx: 2.334
Buffer MAC Count: 0
Verified Msg Count: 0
Invalid MAC Count: 0

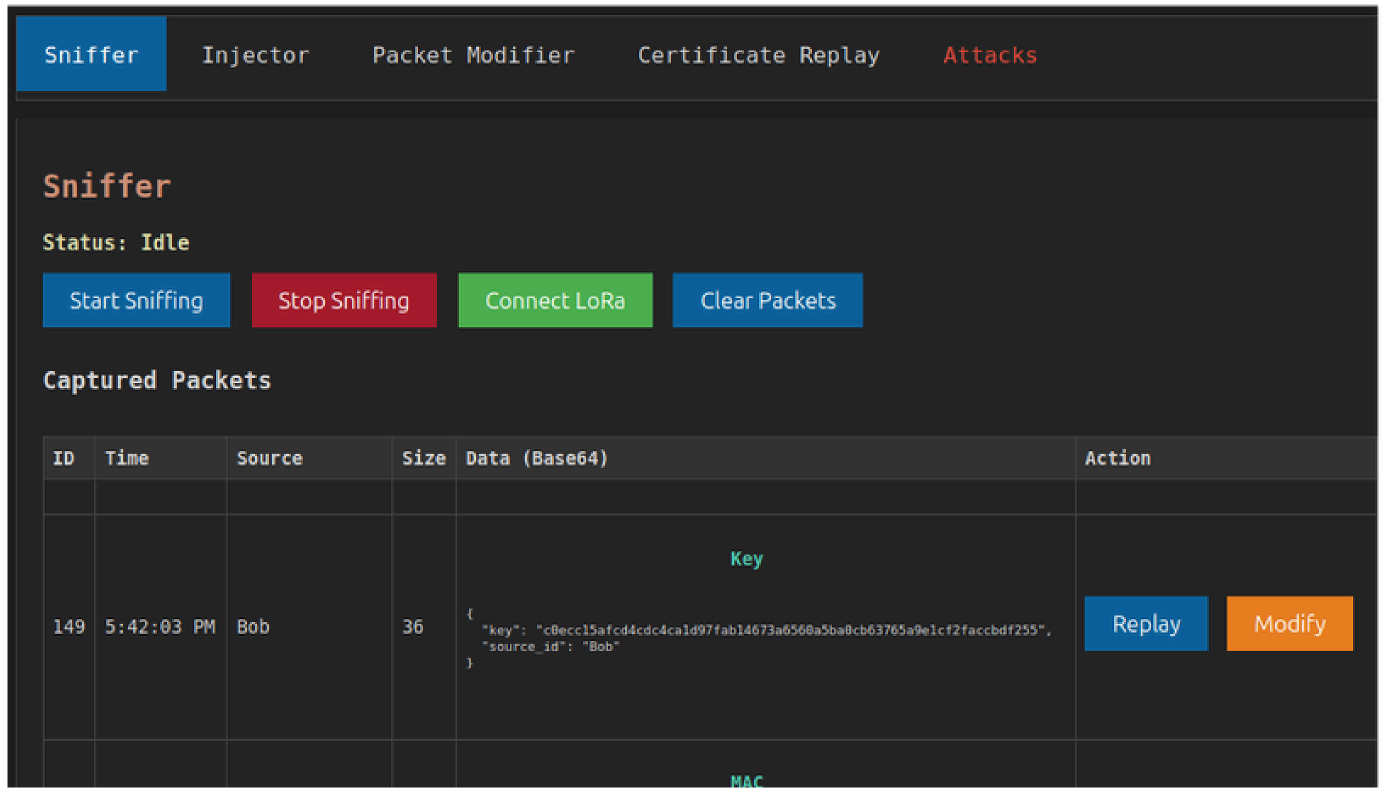
SECURITY VALIDATION AND ATTACK ANALYSIS

RED TEAM FRAMEWORK

- EVIL-VDES: OFFENSIVE MODULE OVER LORA
- SNIFFING & REASSEMBLY

ATTACK VECTORS

- REPLAY & MASQUERADING
- PACKET FORGING
- DOS & JAMMING



Sniffer

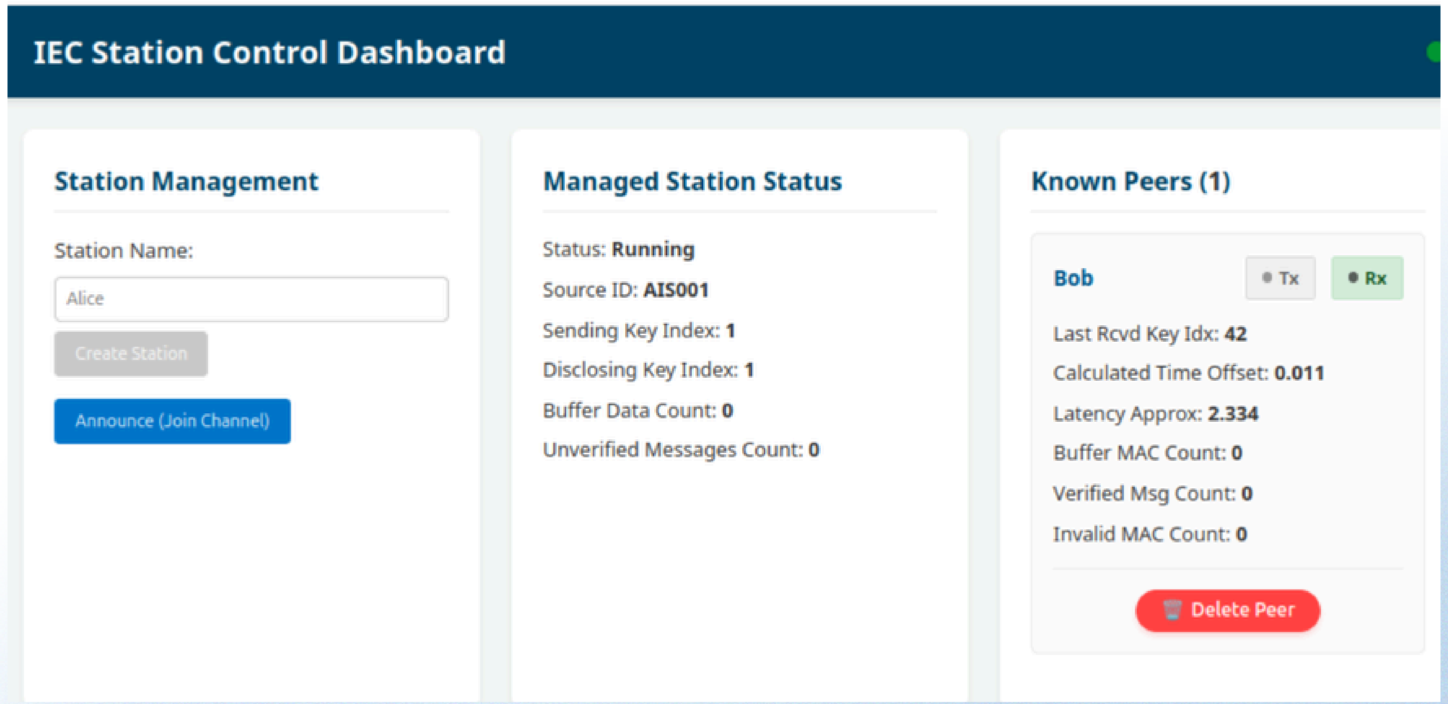
Status: Idle

Captured Packets

ID	Time	Source	Size	Data (Base64)	Action

149	5:42:03 PM	Bob	36	Key
				{
				"key": "c0ecc15afcd4cdc4ca1d97fab14673a6560a5ba0cb63765a9e1cf2faccbdf255",
				"source_id": "Bob"
				}
					Replay Modify

				MAC	



IEC Station Control Dashboard

Station Management

Station Name: Alice

Managed Station Status

Status: **Running**

Source ID: **AIS001**

Sending Key Index: **1**

Disclosing Key Index: **1**

Buffer Data Count: **0**

Unverified Messages Count: **0**

Known Peers (1)

Bob

Last Rcvd Key Idx: **42**

Calculated Time Offset: **0.011**

Latency Approx: **2.334**

Buffer MAC Count: **0**

Verified Msg Count: **0**

Invalid MAC Count: **0**

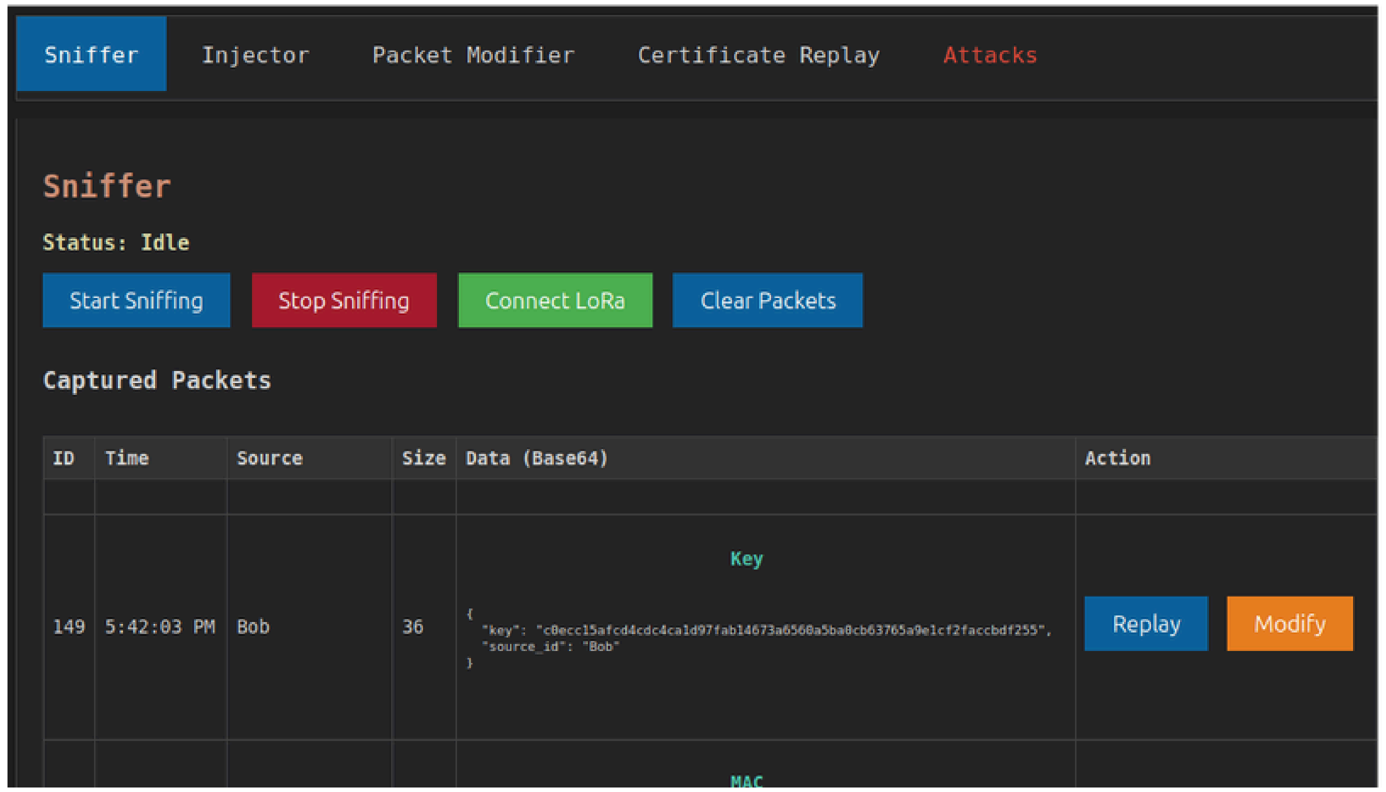
SECURITY VALIDATION AND ATTACK ANALYSIS

RED TEAM FRAMEWORK

- EVIL-VDES: OFFENSIVE MODULE OVER LORA
- SNIFFING & REASSEMBLY

ATTACK VECTORS

- REPLAY & MASQUERADING
- PACKET FORGING
- DOS & JAMMING



Sniffer

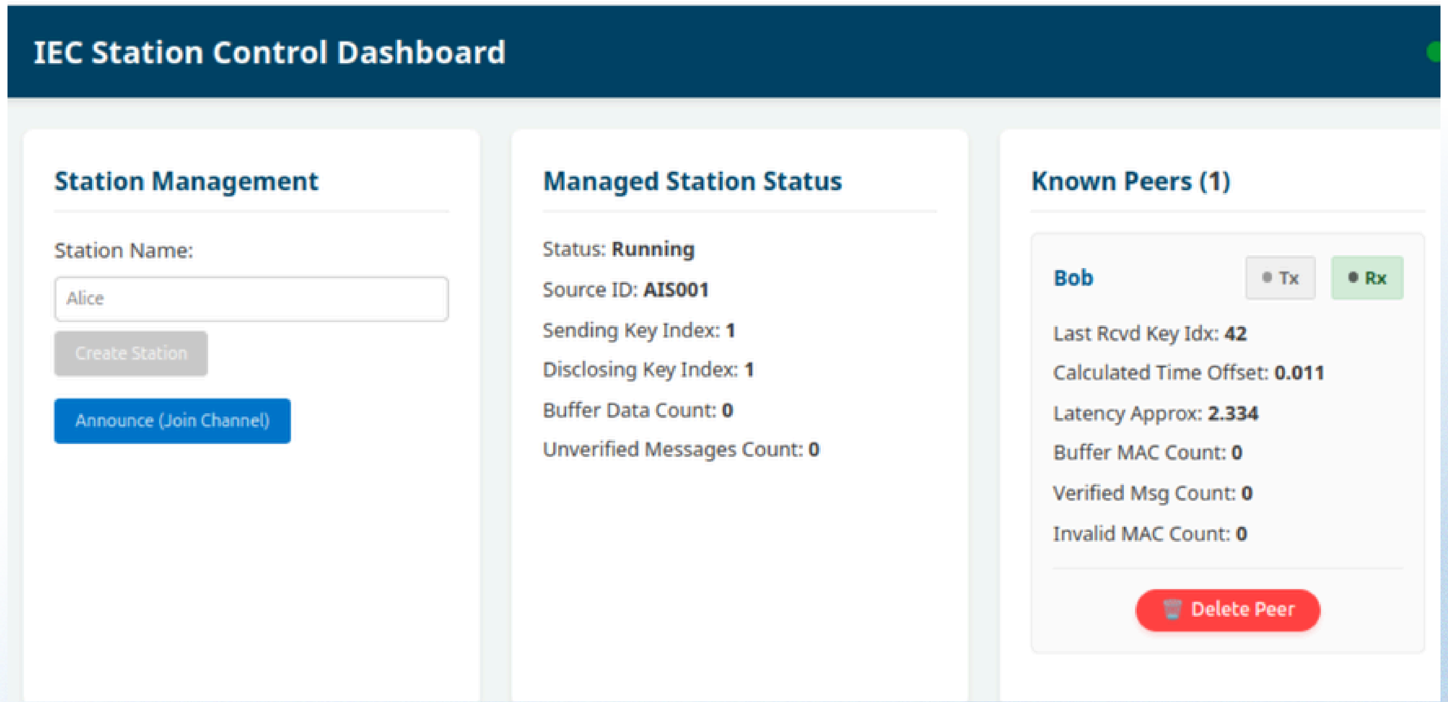
Status: Idle

Captured Packets

ID	Time	Source	Size	Data (Base64)	Action

149	5:42:03 PM	Bob	36	Key
				{
				"key": "c0ecc15afcd4cdc4ca1d97fab14673a6560a5ba0cb63765a9e1cf2faccbdf255",
				"source_id": "Bob"
				}
					Replay Modify

				MAC	



IEC Station Control Dashboard

Station Management

Station Name: Alice

Managed Station Status

Status: **Running**
Source ID: **AIS001**
Sending Key Index: **1**
Disclosing Key Index: **1**
Buffer Data Count: **0**
Unverified Messages Count: **0**

Known Peers (1)

Bob

Last Rcvd Key Idx: **42**
Calculated Time Offset: **0.011**
Latency Approx: **2.334**
Buffer MAC Count: **0**
Verified Msg Count: **0**
Invalid MAC Count: **0**

LINKS & THANKS

- IALA GUIDELINE VDES AUTHENTICATION TECHNIQUES (IALA G1192)
- ION PAPER : A PRAGMATIC APPROACH TO VDES AUTHENTICATION
- PAPER :TESLA PROTOCOL
- SECOM STANDARD (IEC 63173-2)
- IEC 63173-2
- IMO
- ITU

Thank you for your attention,
and thanks to CNIT for the opportunity
to carry out this thesis

cnit consorzio nazionale
interuniversitario
per le telecomunicazioni