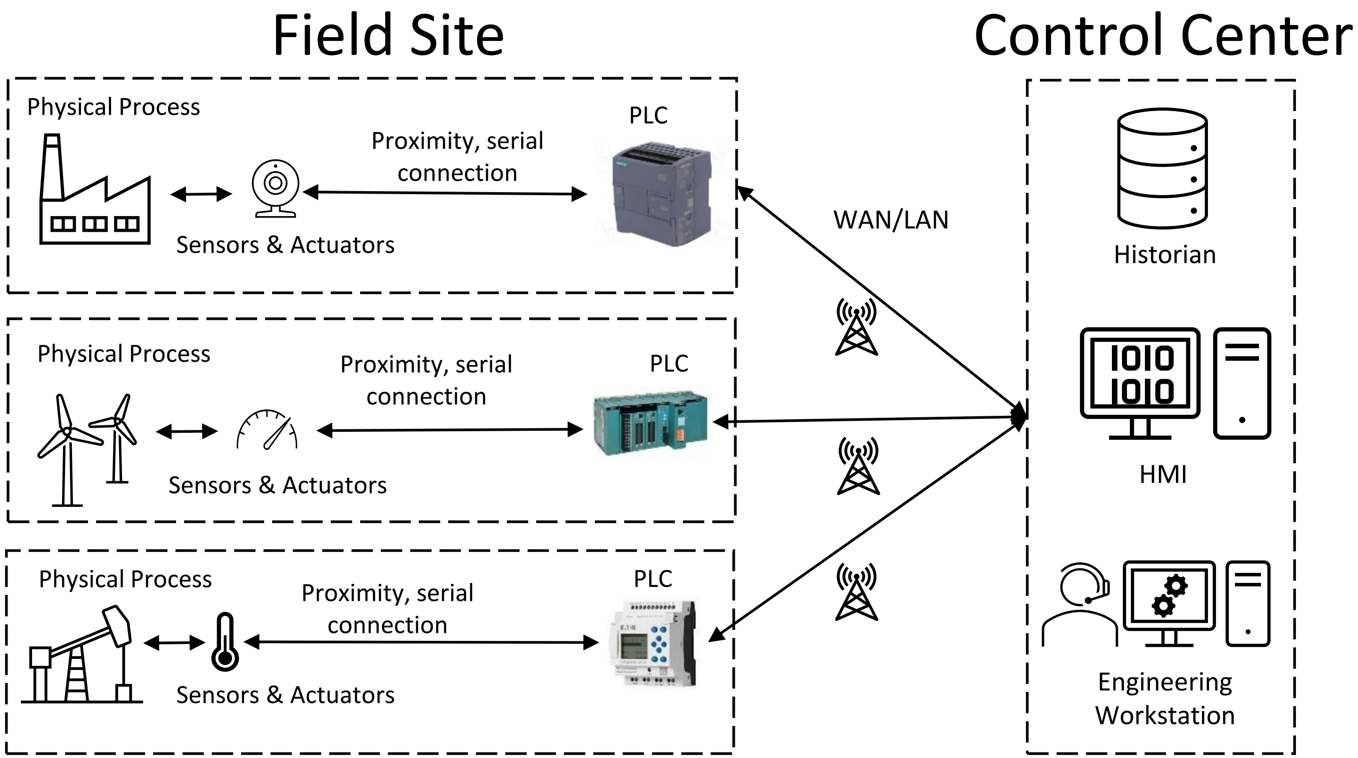
PREE: Heuristic Builder for Reverse Engineering of Network Protocols in Industrial Control Systems

Syed Ali Qasim, Wooyeon Jo, Irfan Ahmed
(qasimsa, jow, iahmed3) @vcu.edu
Virginia Commonwealth University
Richmond, VA

Industrial Control Systems



Attacks on ICS and Forensic Challenges

The New York Times

NEWS ANALYSIS

A Silent Attack, but Not a Subtle One

Iran's Natanz nuclear enrichment site is the focus of speculation about the int
target of a broad and unsubtle cyberattack. Majid Saeedi/Getty Images

Cyberattack on Critical Infrastructure: Russia and the Ukrainian Power Grid Attacks

OCTOBER 11, 2017 // AUTHORS: DONGHUI PARK, MICHAEL WALSTROM

Image credit: NASA's Marshall Space Flight Center

Late

New h
Jackso

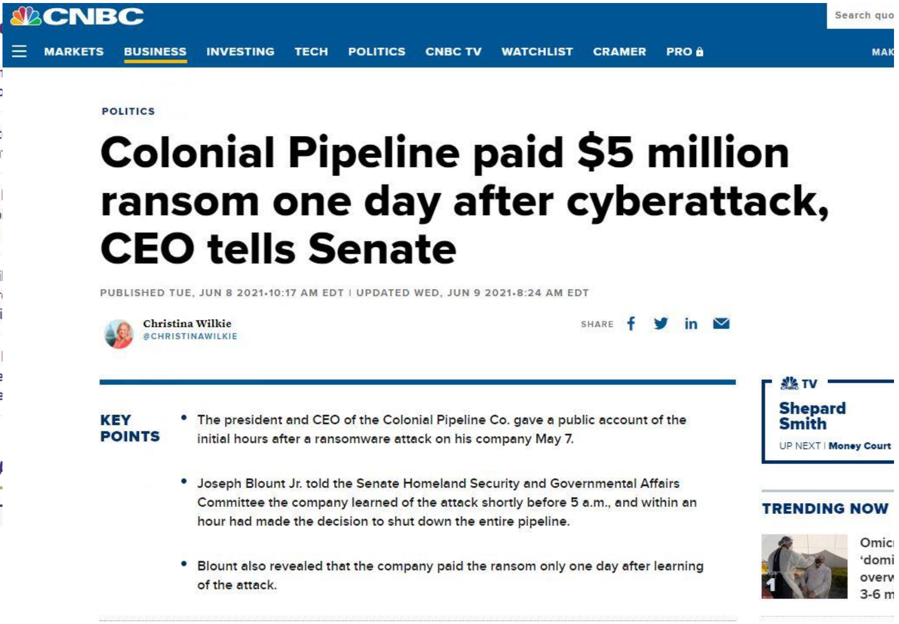
Christ
Volun

Scott l
consp
// LSE

Radhil
Anthr
Menti

Scott l
deepe
Matte

RELA



POLITICS

Colonial Pipeline paid $5 million ransom one day after cyberattack, CEO tells Senate

PUBLISHED TUE, JUN 8 2021•10:17 AM EDT | UPDATED WED, JUN 9 2021•8:24 AM EDT

Christina Wilkie
@CHRISTINAWILKIE

SHARE

KEY POINTS

- The president and CEO of the Colonial Pipeline Co. gave a public account of the initial hours after a ransomware attack on his company May 7.
- Joseph Blount Jr. told the Senate Homeland Security and Governmental Affairs Committee the company learned of the attack shortly before 5 a.m., and within an hour had made the decision to shut down the entire pipeline.
- Blount also revealed that the company paid the ransom only one day after learning of the attack.

TV
Shepard Smith
UP NEXT | Money Court

TRENDING NOW

Omic
'domi
overv
3-6 m

The proprietary nature of ICS protocols presents significant challenges for the security and forensic analysis of PLCs

Current Methods for Protocol Reverse Engineering & their Limitations

Current methods include:

- Manual Reverse Engineering
- Automatic Reverse Engineering (Binary and Network)

Limitations

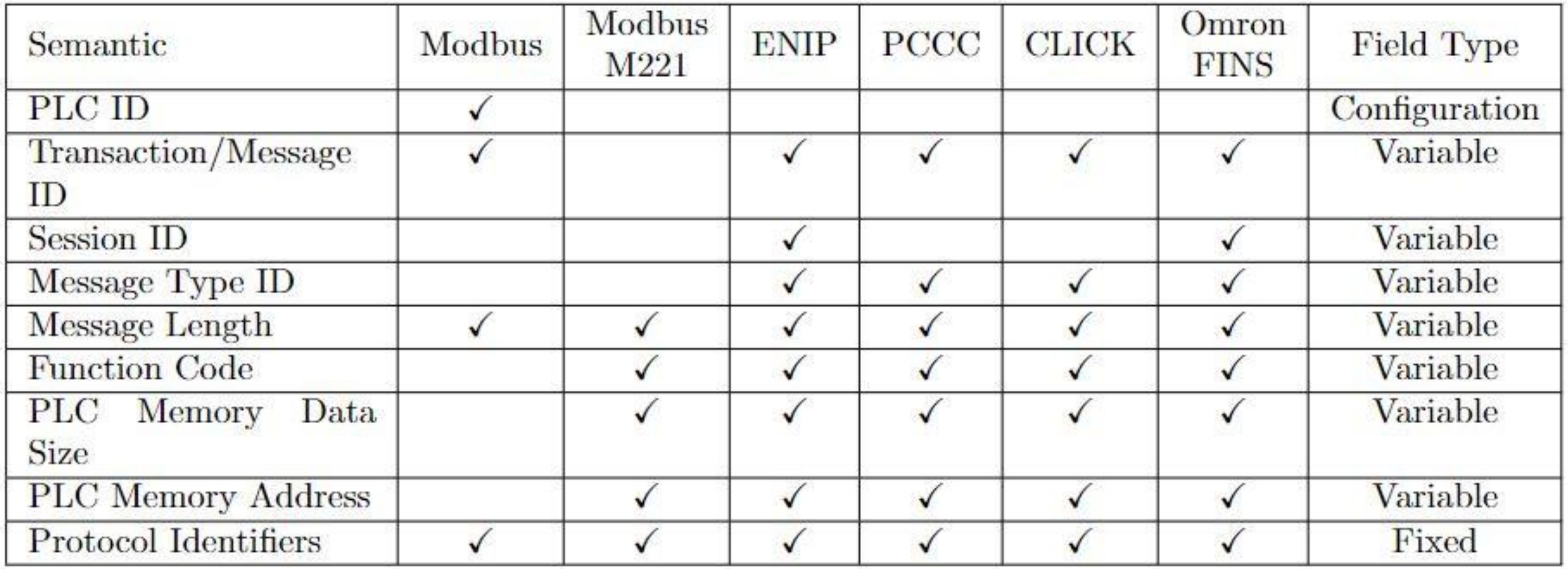
- Manual Analysis: Large Data Volume, Time consuming, Unreadable Binary Messages
- Binary Analysis: Program execution and memory usage, executables files
- Network Analysis: Require Large Data Volumes, High False positive rates

Discovering Common Ground: Unveiling Shared Fields in ICS Protocols

ICS protocol field categories

- Configurational Fields
- Fixed Fields
- Variable Fields

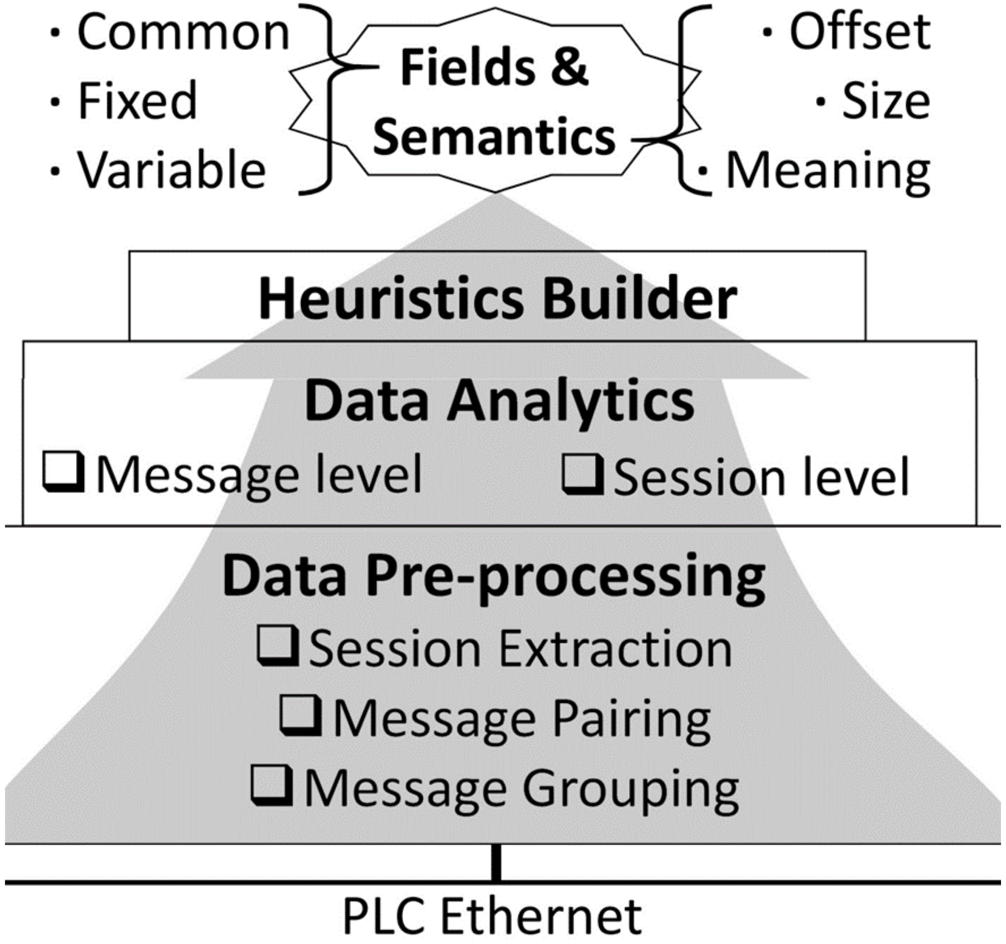
Common fields in different ICS protocols



Semantic	Modbus	Modbus M221	ENIP	PCCC	CLICK	Omron FINS	Field Type
PLC ID	✓						Configuration
Transaction/Message ID	✓		✓	✓	✓	✓	Variable
Session ID			✓			✓	Variable
Message Type ID			✓	✓	✓	✓	Variable
Message Length	✓	✓	✓	✓	✓	✓	Variable
Function Code		✓	✓	✓	✓	✓	Variable
PLC Memory Data Size		✓	✓	✓	✓	✓	Variable
PLC Memory Address		✓	✓	✓	✓	✓	Variable
Protocol Identifiers	✓	✓	✓	✓	✓	✓	Fixed

Protocol Reverse Engineering Engine (PREE)

- A Heuristic builder for ICS protocols
- Utilizes ICS protocol knowledge to create heuristics for message fields.
- Analyzes network dumps at message and session levels
- Provides data analysis functions for heuristic building



PREE Architecture: Data Pre-processing

Session Extraction

- Separates sessions using four-tuple: source IP, source port, destination IP, destination port

Message Pairing

- Pairs request and response messages and maintains the sequence

Message Grouping

- Groups similar messages based on payload length or total size

PREE Architecture: Data Analytics

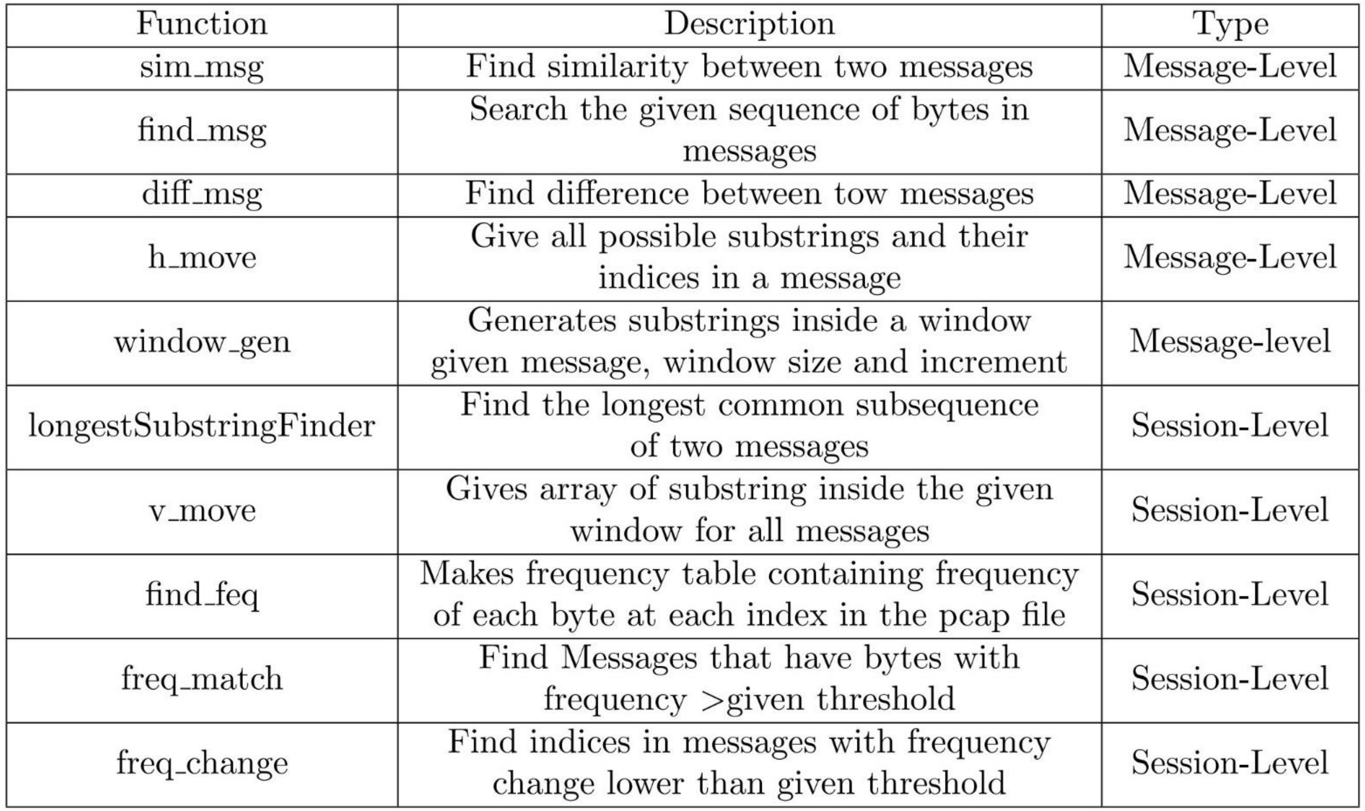
Message-Level Analysis

- Certain protocol fields, such as “Length field” can be identified using information within the message

Session-Level Analysis

- Focuses on session-wide patterns in protocol fields.

Summary of PREE data analytics functionalities



Function	Description	Type
sim_msg	Find similarity between two messages	Message-Level
find_msg	Search the given sequence of bytes in messages	Message-Level
diff_msg	Find difference between tow messages	Message-Level
h_move	Give all possible substrings and their indices in a message	Message-Level
window_gen	Generates substrings inside a window given message, window size and increment	Message-level
longestSubstringFinder	Find the longest common subsequence of two messages	Session-Level
v_move	Gives array of substring inside the given window for all messages	Session-Level
find_feq	Makes frequency table containing frequency of each byte at each index in the pcap file	Session-Level
freq_match	Find Messages that have bytes with frequency >given threshold	Session-Level
freq_change	Find indices in messages with frequency change lower than given threshold	Session-Level

Finding Configuration and Fixed Fields

Finding Configuration Fields:

- No heuristics required
- Use "find_msg" function in PREE
- Takes target sequence of bytes (known configuration field value)
- Returns location/index in all messages of a session if found

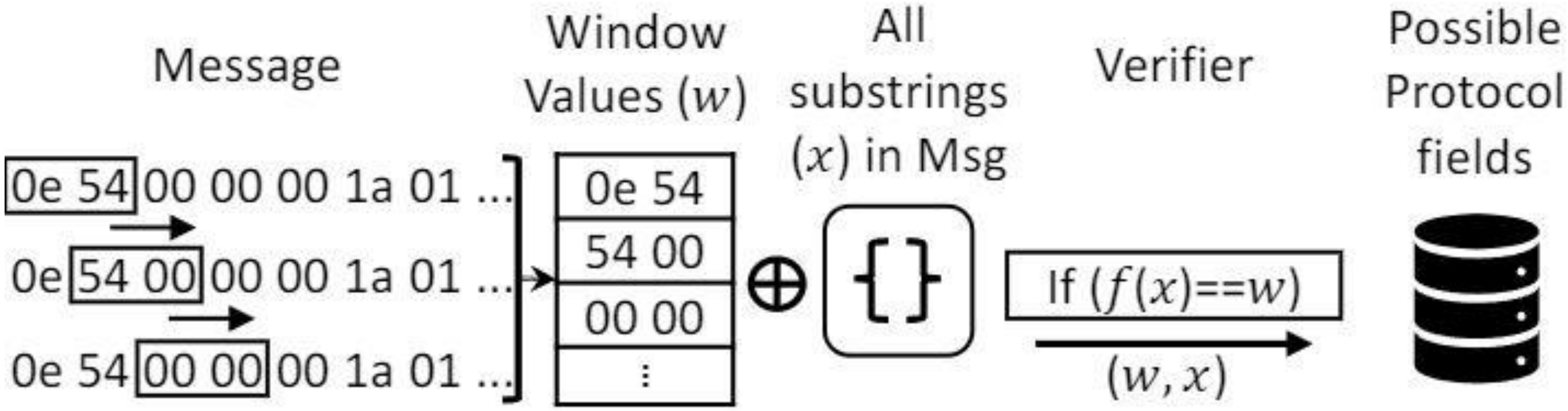
Finding Fixed Fields:

- Use "find_feq" function in PREE
- Generates frequency table of values across message indices in a session.
- Fixed fields found where frequency is 100% (value stays the same)

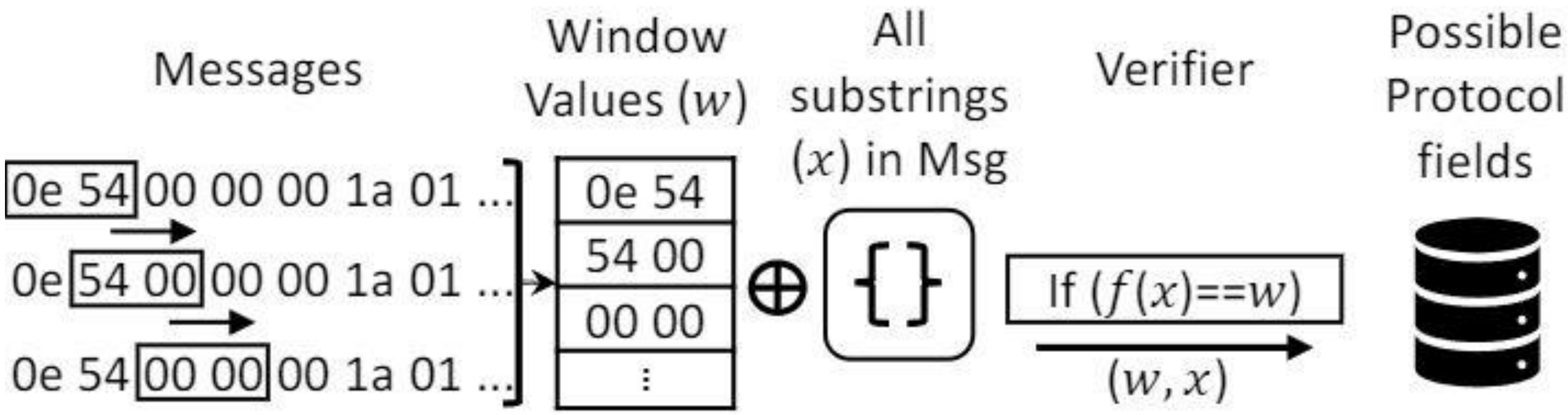
Finding Variable Fields: Rolling Window Technique

Rolling Window:

- Sliding window of varying sizes (1, 2, ..., n bytes) over the message
- Applies user-defined function to all substrings
- Potential fields selected if consistently appearing across similar messages



Finding Varaiable Fields: Rolling Window Techinque



Length field:

- User provides function f(x) to calculate payload length
- Window location marked as a length field if value matches output of f(x)

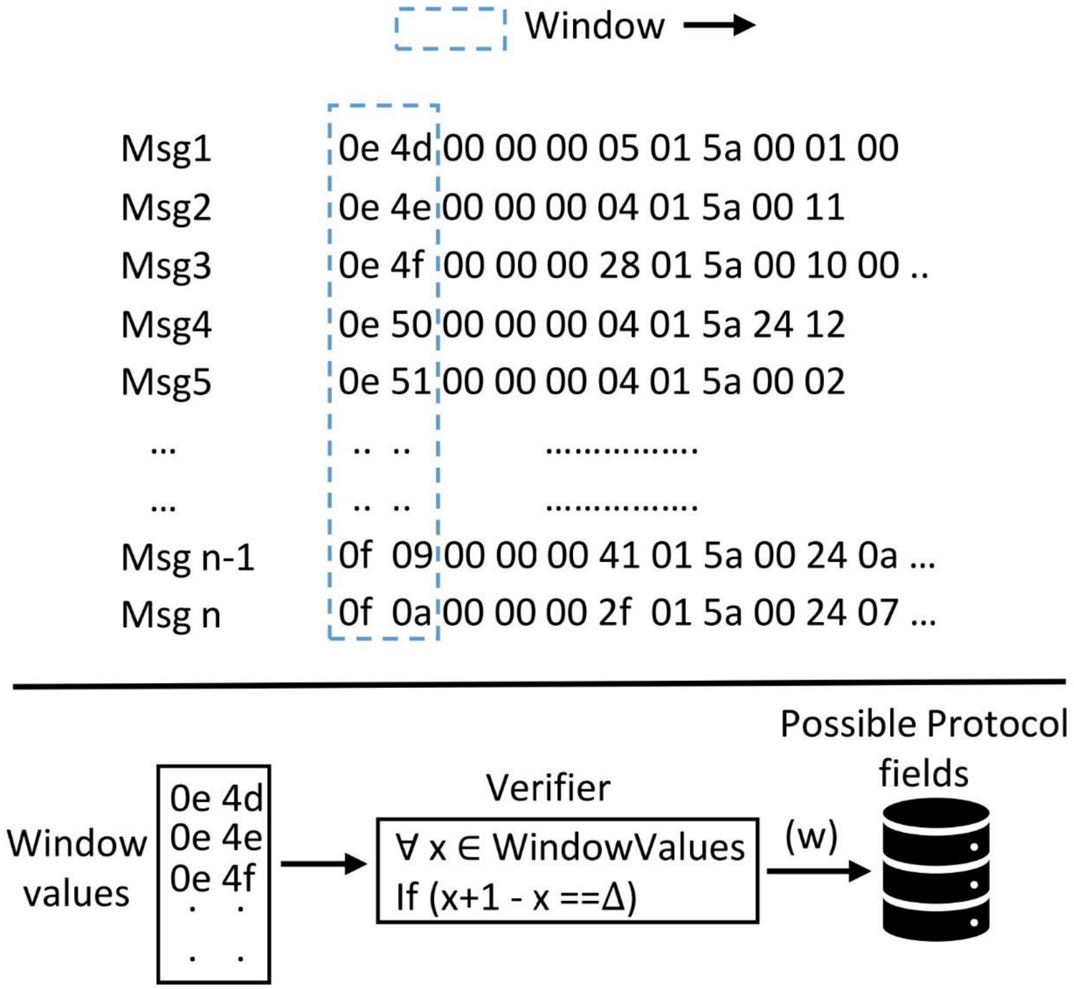
Checksum field:

- User provides potential checksum function
- Rolling window technique identifies the location of the checksum field

Finding Variable Fields: Vertical Window Technique

Vertical Window:

- Moves a window of varying sizes over all messages in a session
- Checks if user-defined function f(y) = y+1 for consecutive message pairs
- Window location labeled a potential protocol field based on f(x)



```
Msg1      0e 4d 00 00 00 05 01 5a 00 01 00
Msg2      0e 4e 00 00 00 04 01 5a 00 11
Msg3      0e 4f 00 00 00 28 01 5a 00 10 00 ..
Msg4      0e 50 00 00 00 04 01 5a 24 12
Msg5      0e 51 00 00 00 04 01 5a 00 02
…         .. ..          …………….
…         .. ..          …………….
Msg n-1   0f 09 00 00 00 41 01 5a 00 24 0a …
Msg n     0f 0a 00 00 00 2f 01 5a 00 24 07 …
```

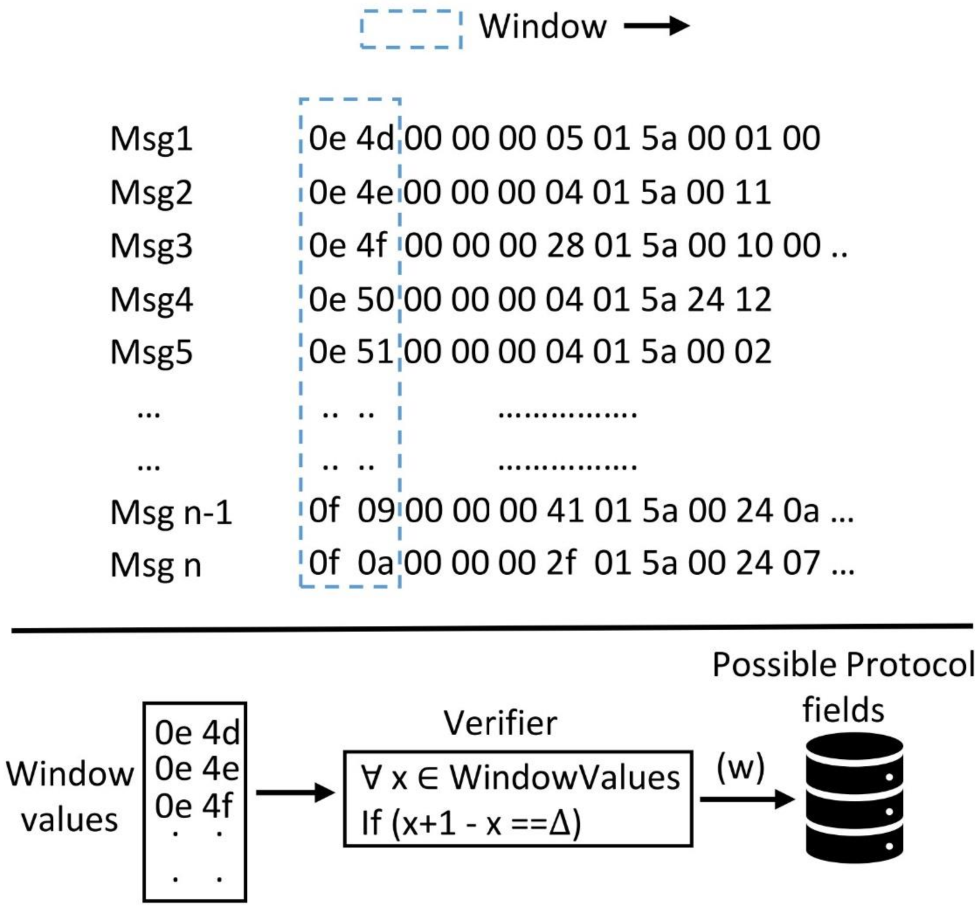
Finding Variable Fields: Vertical Window Technique

Transaction ID:

- Increases constantly with each new message
- Define f(x) to add a fixed number to x
- Sliding window represents potential "Transaction ID"

PLC Memory Address:

- Address changes by the size of data written/read in consecutive messages
- Use f(x) to add current memory address and data size
- Vertical window identifies "PLC Memory Address"



```
Msg1      0e 4d 00 00 00 05 01 5a 00 01 00
Msg2      0e 4e 00 00 00 04 01 5a 00 11
Msg3      0e 4f 00 00 00 28 01 5a 00 10 00 ..
Msg4      0e 50 00 00 00 04 01 5a 24 12
Msg5      0e 51 00 00 00 04 01 5a 00 02
 ...      ..  ..       ...............
 ...      ..  ..       ...............
Msg n-1   0f  09 00 00 00 41 01 5a 00 24 0a ...
Msg n     0f  0a 00 00 00 2f  01 5a 00 24 07 ...
```

Finding Variable Fields: Frequency Table Technique

Frequency Table:

- Identifies variable fields without a specific pattern
- Stores frequency and values of each byte at each index in a session

Frequency Table

1	2	3	4	.	.	n
‘0e’:52 ‘0f’:60	‘4d’:1 ‘4e’:1 ‘4f’:1 …..	‘00’:102	‘00’:102	..	…	….

Finding Variable Fields: Frequency Table Technique

Session ID:

- Exchanged in the beginning and stays constant afterwards
- Query frequency table for indices with limited changes
- Search bytes in initial messages to find "Session ID"

Function Code:

- Limited set of codes in requests and responses
- Query frequency table for limited variance in request messages and constant values in response messages
- Indices may indicate "Function Code" in ICS protocol

Message Type ID:

- Unique values in request and response messages
- Create separate frequency tables for request and response messages
- Compare bytes with 100% frequency in each table to find "Message Type ID"

Evaluation Metrics for PREE

Coverage:

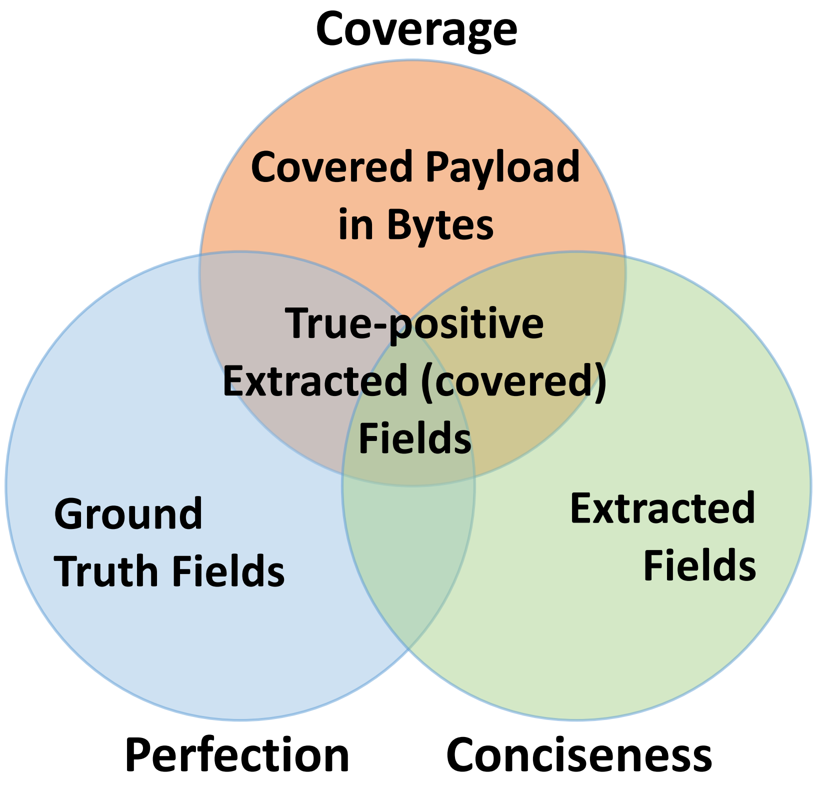
- Percentage of messages covered by PREE as protocol fields.

Perfection:

- Quality of perfect extraction of existing ground truth fields.

Conciseness:

- How efficiently we are able to extract the relevant ground truth fields.



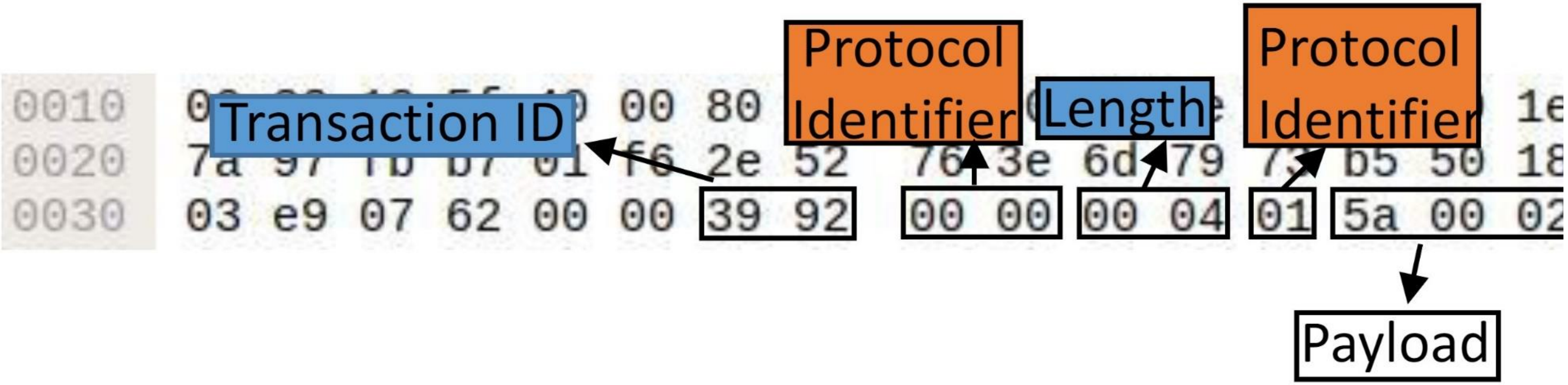
$$Conciseness = \frac{\#\ of\ extracted\ ground\ truth\ fields}{\#\ of\ extracted\ fields}$$

$$Coverage = \frac{\#\ of\ labeled\ bytes}{\#\ of\ extracted\ bytes}$$

$$Perfection = \frac{\#\ of\ extracted\ ground\ truth\ fields}{\#\ of\ total\ ground\ truth\ fields}$$

PREE Evaluation: Modbus

Fields Identified In Modbus Protocol



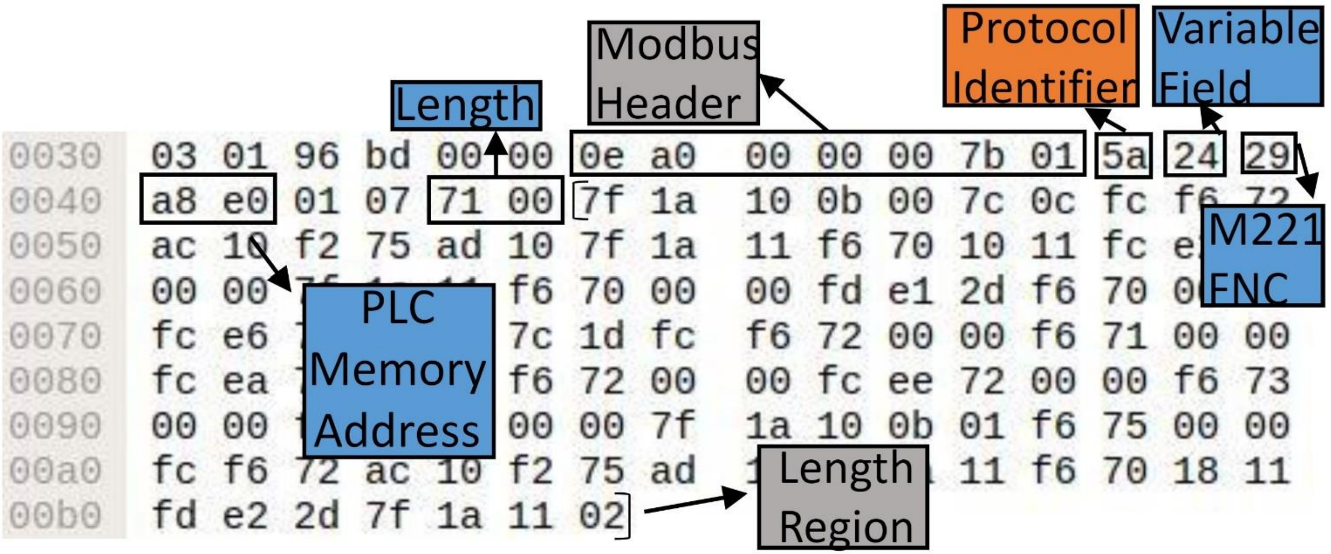
```
0030   03 e9 07 62 00 00 39 92 00 00 00 04 01 5a 00 02
```

Comparison of PREE and Ground Truth For Modbus

Field	PREE Location	Ground Truth Location	PREE Semantic	Ground Truth Semantic	#PREE types	# Ground Truth types
1	1-2	1-2	Transaction ID	Transaction ID	1	1
2	5-6	5-6	Length	Length	1	1
3	3-4	3-4	Protocol ID	Protocol ID	1	1
4	7	7	Protocol ID	Protocol ID	1	1

PREE Evaluation: UMAS

Fields Identified In UMAS Protocol



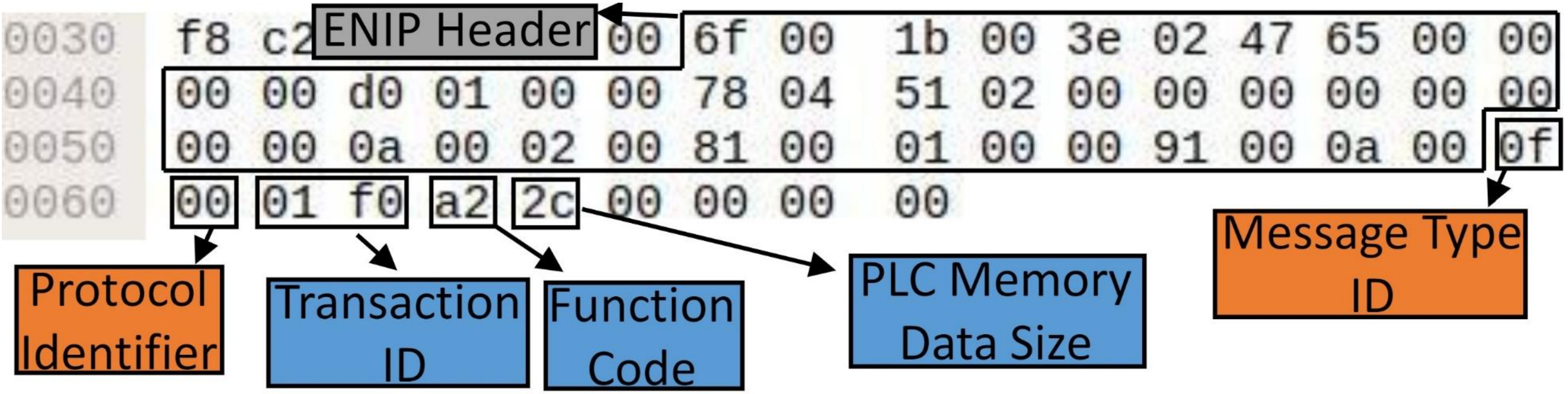
```
0030  03 01 96 bd 00 00 0e a0  00 00 00 7b 01 5a 24 29
0040  a8 e0 01 07 71 00 7f 1a  10 0b 00 7c 0c fc f6 72
0050  ac 10 f2 75 ad 10 7f 1a  11 f6 70 10 11 fc e
0060  00 00          f6 70 00  00 fd e1 2d f6 70 0
0070  fc e6          7c 1d fc  f6 72 00 00 f6 71 00 00
0080  fc ea          f6 72 00  00 fc ee 72 00 00 f6 73
0090  00 00          00 00 7f  1a 10 0b 01 f6 75 00 00
00a0  fc f6 72 ac 10 f2 75 ad  1           11 f6 70 18 11
00b0  fd e2 2d 7f 1a 11 02
```

Comparison of PREE and Ground Truth for UMAS

Field	PREE Location	Ground Truth Location	PREE Semantic	Ground Truth Semantic	#PREE types	# Ground Truth types
1	1	1	Protocol ID	Protocol ID	1	1
2	3	3	Function Code	Function Code	1	1
3	4-5	4-5	PLC Memory Address	PLC Memory Address	1	1
4	8-9	8-9	Length	PLC Memory Data Size	1	1

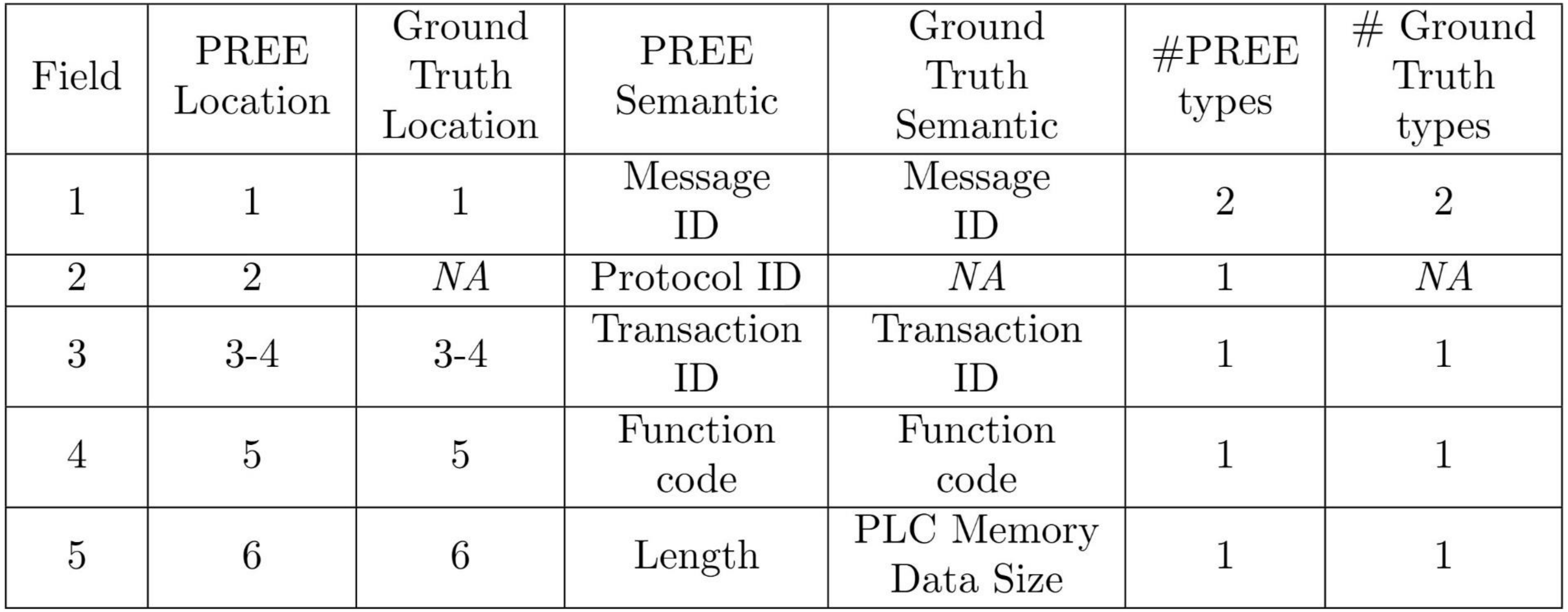
PREE Evaluation: PCCC

Fields Identified In PCCC Protocol



```
0030   f8 c2             00  6f 00   1b 00 3e 02 47 65 00 00
0040   00 00 d0 01 00 00 78 04   51 02 00 00 00 00 00 00
0050   00 00 0a 00 02 00 81 00   01 00 00 91 00 0a 00 0f
0060   00 01 f0 a2 2c 00 00 00   00
```

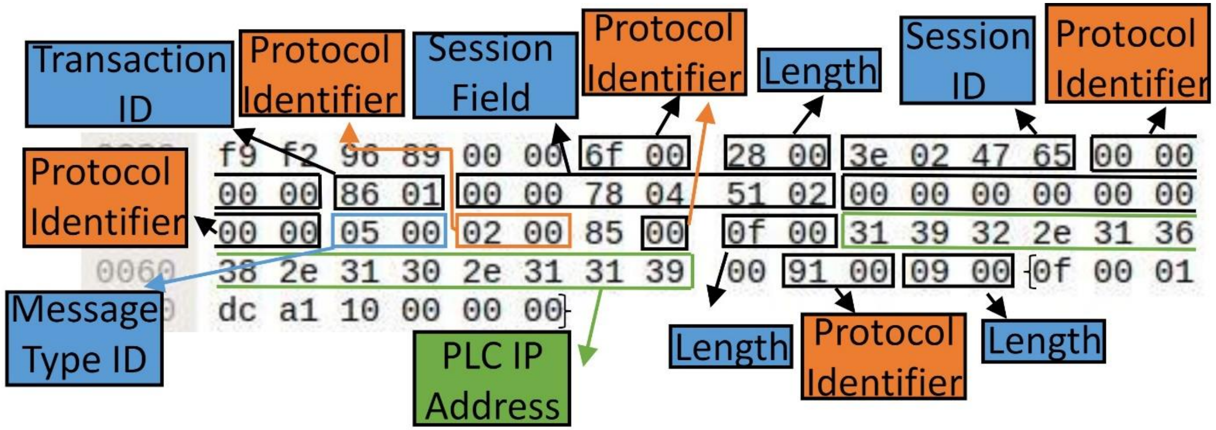
Comparison of PREE and Ground Truth for PCCC



Field	PREE Location	Ground Truth Location	PREE Semantic	Ground Truth Semantic	#PREE types	# Ground Truth types
1	1	1	Message ID	Message ID	2	2
2	2	*NA*	Protocol ID	*NA*	1	*NA*
3	3-4	3-4	Transaction ID	Transaction ID	1	1
4	5	5	Function code	Function code	1	1
5	6	6	Length	PLC Memory Data Size	1	1

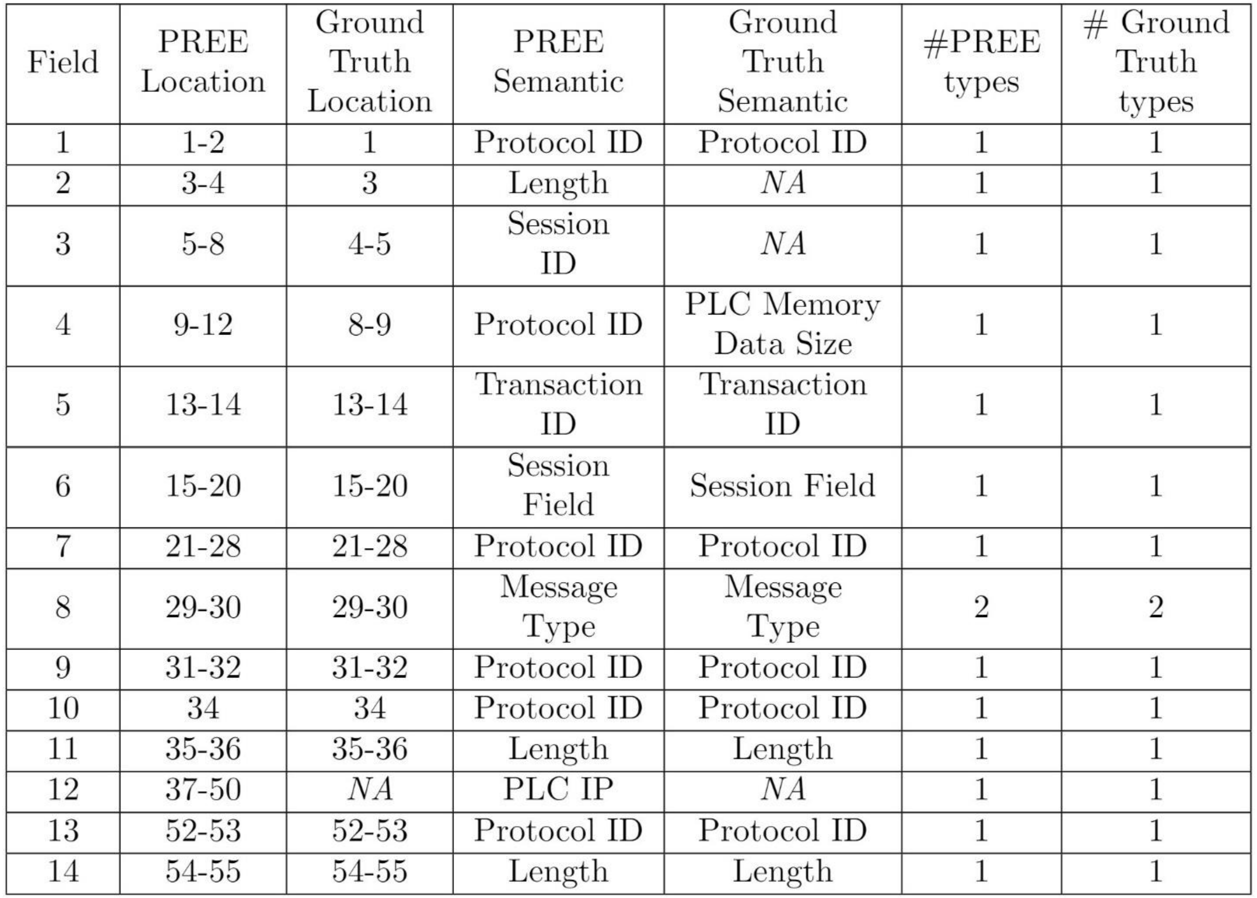
PREE Evaluation: ENIP

Fields Identified in ENIP Protocol



```
      f9 f2 96 89 00 00 6f 00  28 00 3e 02 47 65 00 00
      00 00 86 01 00 00 78 04  51 02 00 00 00 00 00 00
      00 00 05 00 02 00 85 00  0f 00 31 39 32 2e 31 36
0060  38 2e 31 30 2e 31 31 39  00 91 00 09 00 0f 00 01
      dc a1 10 00 00 00
```

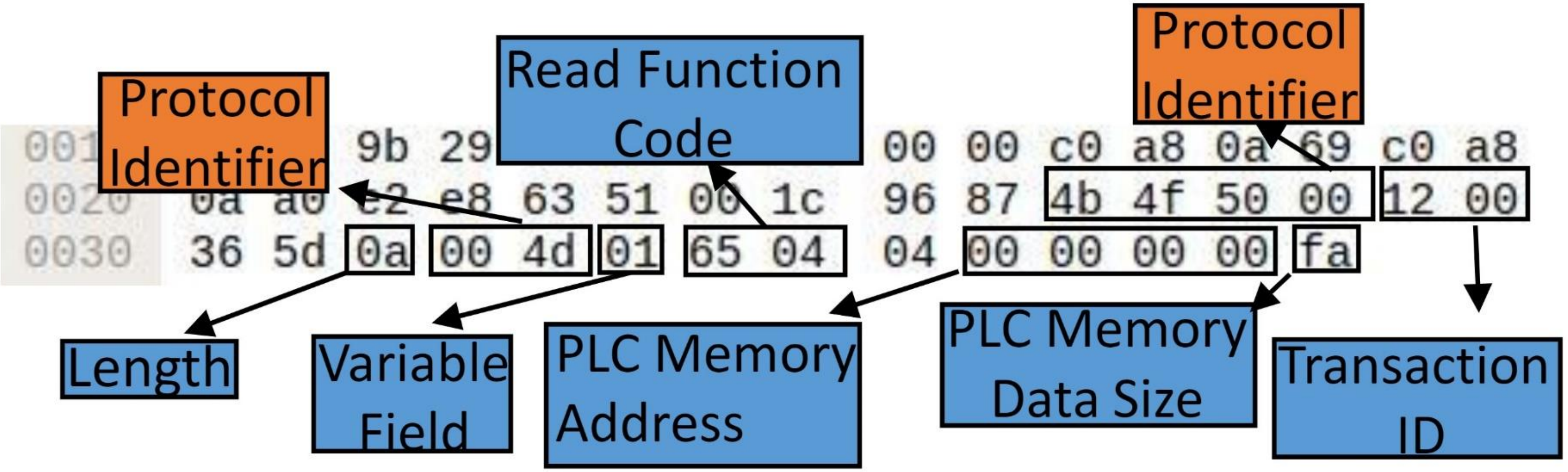
Comparison of PREE and Ground Truth for ENIP



Field	PREE Location	Ground Truth Location	PREE Semantic	Ground Truth Semantic	#PREE types	# Ground Truth types
1	1-2	1	Protocol ID	Protocol ID	1	1
2	3-4	3	Length	*NA*	1	1
3	5-8	4-5	Session ID	*NA*	1	1
4	9-12	8-9	Protocol ID	PLC Memory Data Size	1	1
5	13-14	13-14	Transaction ID	Transaction ID	1	1
6	15-20	15-20	Session Field	Session Field	1	1
7	21-28	21-28	Protocol ID	Protocol ID	1	1
8	29-30	29-30	Message Type	Message Type	2	2
9	31-32	31-32	Protocol ID	Protocol ID	1	1
10	34	34	Protocol ID	Protocol ID	1	1
11	35-36	35-36	Length	Length	1	1
12	37-50	*NA*	PLC IP	*NA*	1	1
13	52-53	52-53	Protocol ID	Protocol ID	1	1
14	54-55	54-55	Length	Length	1	1

PREE Evaluation: CLICK

Fields Identified in CLICK Protocol



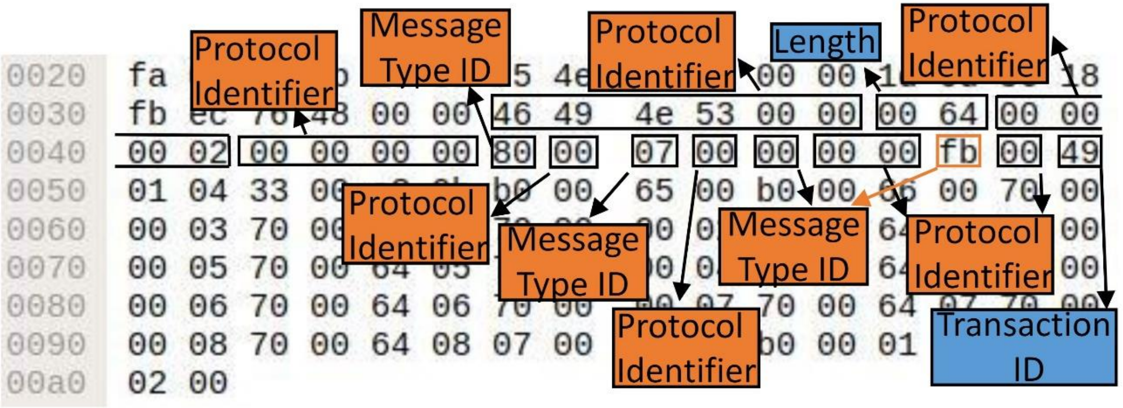
```
001  9b 29  00 00 c0 a8 0a 69 c0 a8
0020 0a a0 e2 e8 63 51 00 1c  96 87 4b 4f 50 00 12 00
0030 36 5d 0a 00 4d 01 65 04 04 00 00 00 00 fa
```

Comparison of PREE and Ground Truth for CLICK

Field	PREE Location	Ground Truth Location	PREE Semantic	Ground Truth Semantic	#PREE types	# Ground Truth types
1	1-4	1-4	Protocol ID	Protocol ID	1	1
2	5-6	5-6	Transaction ID	Transaction ID	1	1
3	9	9	Length	Length	1	1
4	10-11	10-11	Protocol ID	Protocol ID	1	1
5	15	15	PLC Memory Data Size	PLC Memory Data Size	1	1
6	16-19	16-19	PLC Memory Address	PLC Memory Address	1	1
7	20	20	Length	PLC Memory Data Size	1	1

PREE Evaluation: OMRON FINS

Fields Identified in FINS Protocol



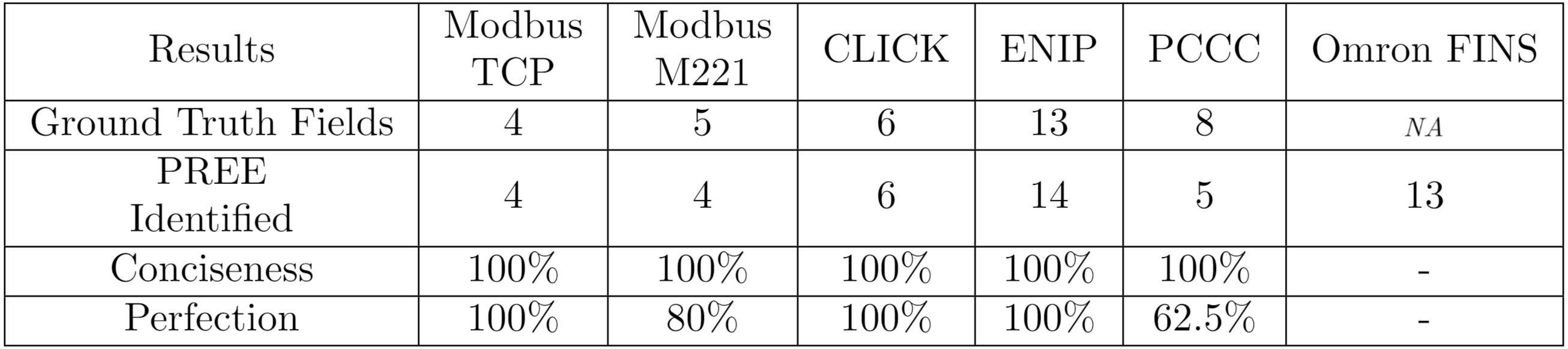
Comparison of PREE and Ground Truth for FINS

Field	PREE Location	Ground Truth Location	PREE Semantic	Ground Truth Semantic	#PREE types	# Ground Truth types
1	1-6	*NA*	Protocol ID	*NA*	1	*NA*
2	7-8	*NA*	Length	*NA*	1	*NA*
3	9-16	*NA*	Length	*NA*	1	*NA*
4	17	*NA*	Message Type ID	*NA*	1	*NA*
5	18	*NA*	Protocol ID	*NA*	1	*NA*
6	19	*NA*	Message Type ID	*NA*	1	*NA*
7	20	*NA*	Protocol ID	*NA*	1	*NA*
8	21	*NA*	Message Type ID	*NA*	1	*NA*
9	22-23	*NA*	Protocol ID	*NA*	1	*NA*
10	24	*NA*	Message Type ID	*NA*	1	*NA*
11	25	*NA*	Protocol ID	*NA*	1	*NA*
12	26	*NA*	Transaction ID	*NA*	1	*NA*

Evaluation Summary

- Five PLCs from four ICS vendors
- Six ICS protocols tested
- Three techniques and seven heuristics used
- Eight protocol fields effectively identified

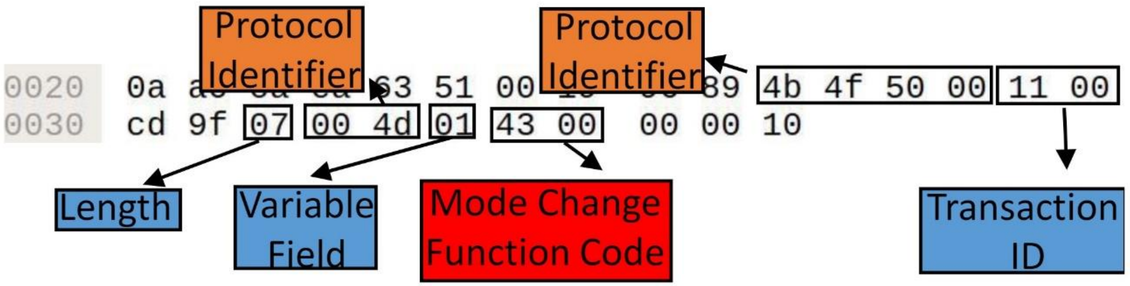
Summary of Fields Identified by PREE



Results	Modbus TCP	Modbus M221	CLICK	ENIP	PCCC	Omron FINS
Ground Truth Fields	4	5	6	13	8	*NA*
PREE Identified	4	4	6	14	5	13
Conciseness	100%	100%	100%	100%	100%	-
Perfection	100%	80%	100%	100%	62.5%	-

PREE Application I: Vulnerability Discovery on Click PLC

PLC Mode Change Message



Adversary Model:

- Adversary inside the ICS network
- Can communicate with the target PLC
- Can sniff communication, initiate connections, and send malicious messages

Attack Implementation:

- Changing the mode of a PLC using engineering software
- Capturing network traffic and analyzing differences
- Identifying messages responsible for switching the PLC from start to stop

PREE Application II: Forensic Analysis of ICS Attacks on Click PLC

Snort Rules

Control Engine Attack

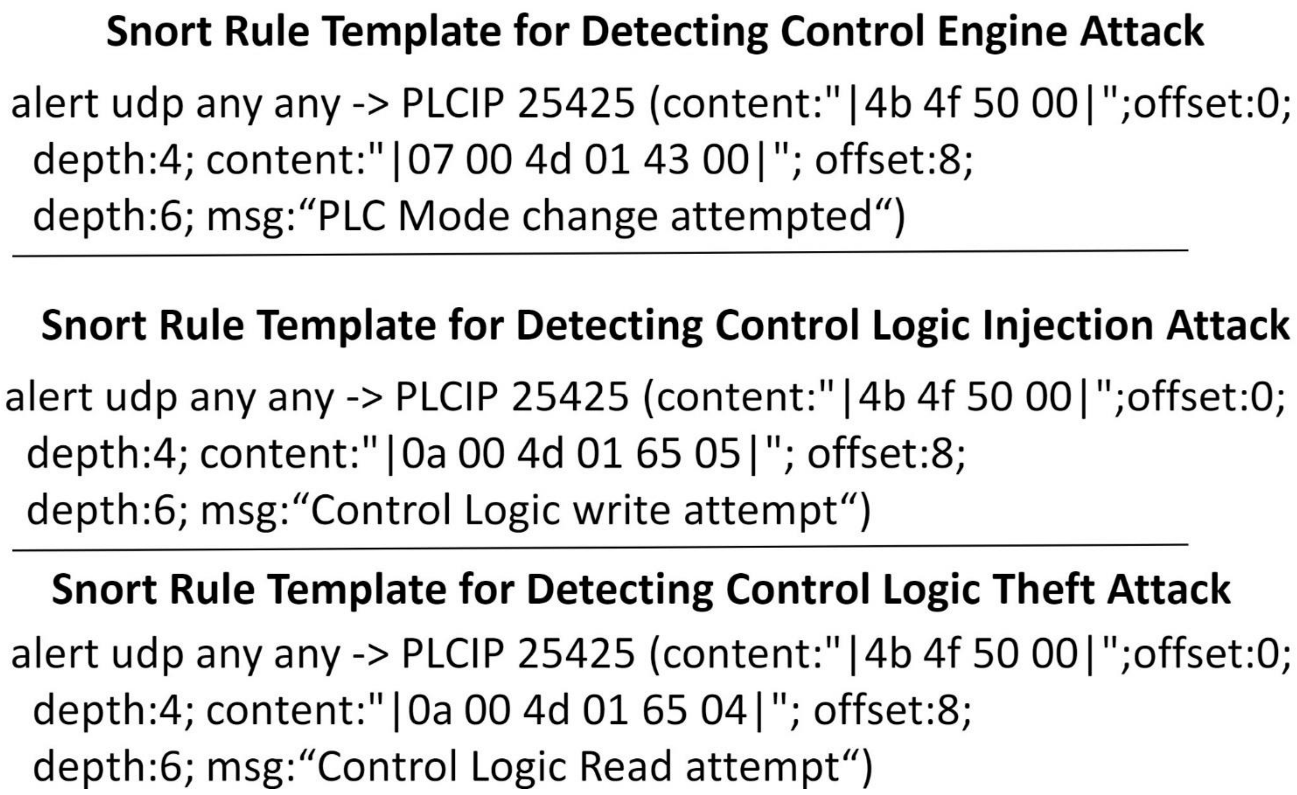
- SNORT rule raises an alert for messages containing the signature of a PLC mode change

Control Logic Injection Attack

- Raises alert when it detects a write request FC ‘05’

Control Logic Theft Attack

- Raises alert when it detects a read request FC ‘04’



Snort Rule Template for Detecting Control Engine Attack

```
alert udp any any -> PLCIP 25425 (content:"|4b 4f 50 00|";offset:0;
 depth:4; content:"|07 00 4d 01 43 00|"; offset:8;
 depth:6; msg:“PLC Mode change attempted“)
```

Snort Rule Template for Detecting Control Logic Injection Attack

```
alert udp any any -> PLCIP 25425 (content:"|4b 4f 50 00|";offset:0;
 depth:4; content:"|0a 00 4d 01 65 05|"; offset:8;
 depth:6; msg:“Control Logic write attempt“)
```

Snort Rule Template for Detecting Control Logic Theft Attack

```
alert udp any any -> PLCIP 25425 (content:"|4b 4f 50 00|";offset:0;
 depth:4; content:"|0a 00 4d 01 65 04|"; offset:8;
 depth:6; msg:“Control Logic Read attempt“)
```

Conclusion

- Developed PREE: a tool for reversing proprietary ICS protocols based on shared common fields
- PREE assists users in creating heuristics for identifying fields in various protocols
- Applied seven heuristics to six protocols (Modbus, UMAS, ENIP, PCCC, CLICK, OMRON FINS) using three techniques
- Successfully identified several common fields in these protocols
- Demonstrated practical applications for investigating 3 different network-based attacks on CLICK PLC

Thank You

Questions?