SCADA Network Forensics of the PCCC Protocol

Saranyan Senthivel, Dr.Irfan Ahmed, Dr. Vassil Roussev

Department of Computer Science
Greater New Orleans Center for Information Assurance
University of New Orleans

GNOCIA
Greater New Orleans Center for Information Assurance

Agenda

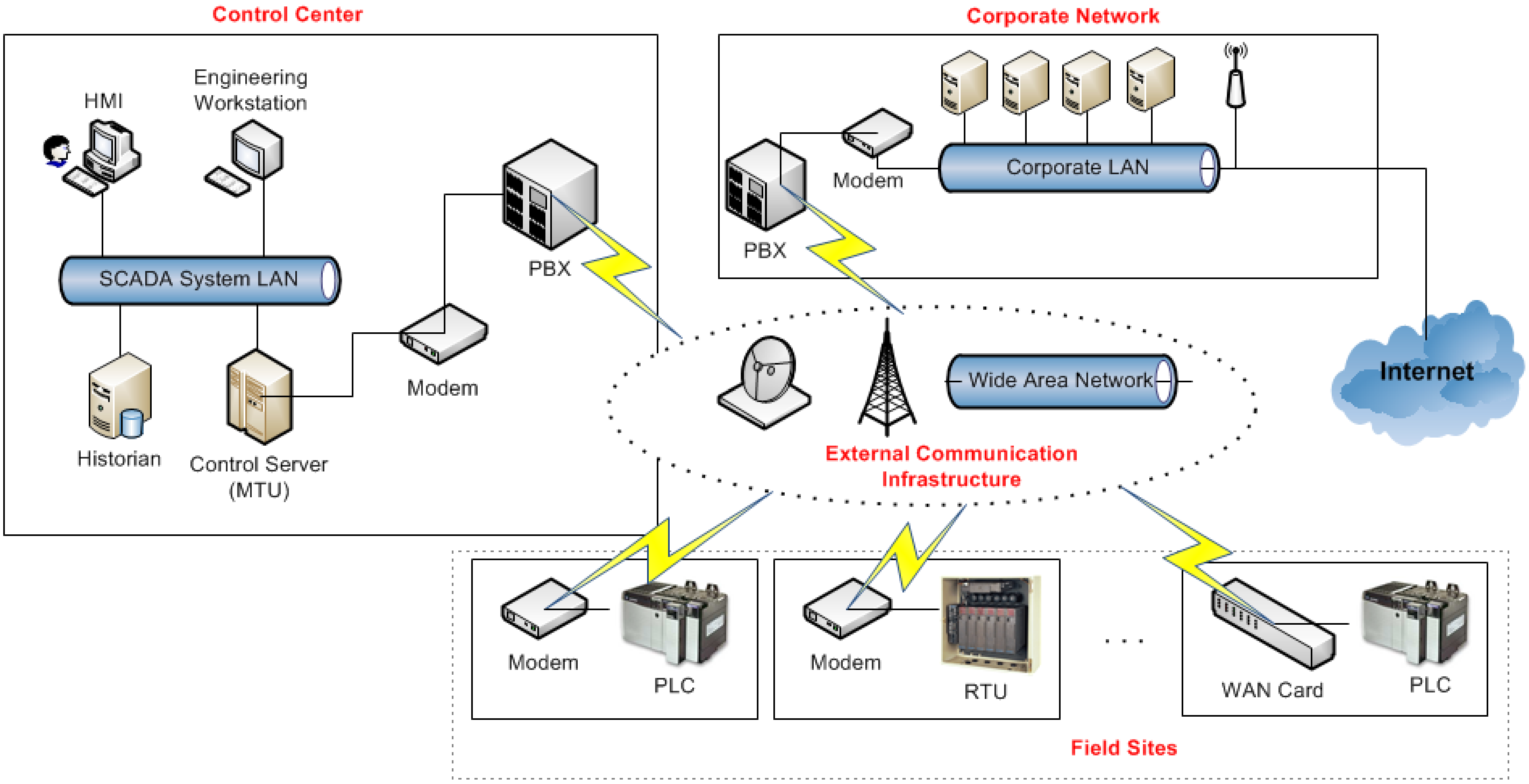
- Insight to SCADA & PCCC
- Implementation
- Finding Digital Artifacts
- Evaluation
- Conclusion

Geographically Dispersed Assets

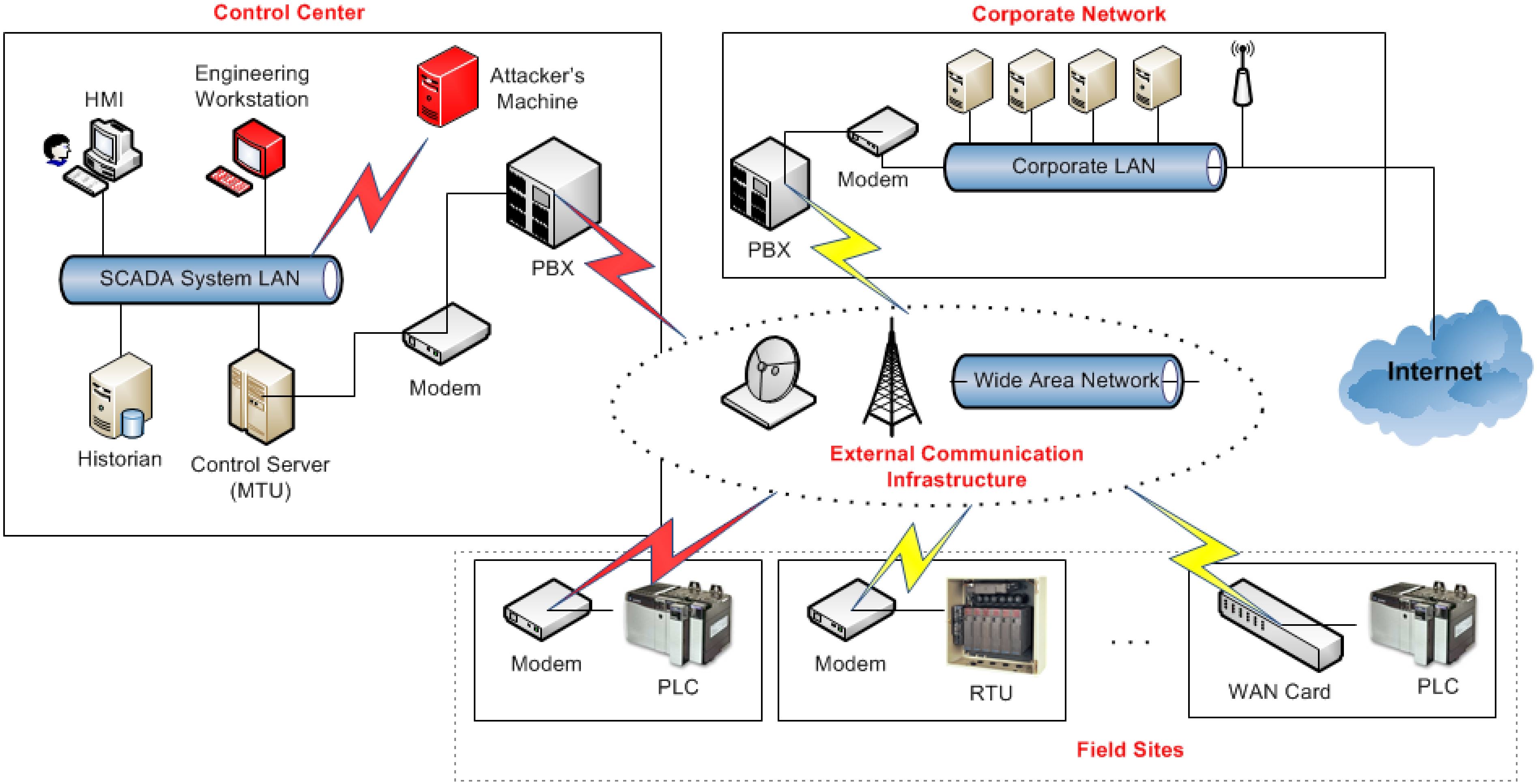
SCADA Systems

- SCADA → Supervisory Control and Data Acquisition
- Are highly distributed systems
- Provides centralized data acquisition, monitoring, and control in real time
- Program PLC's using Ladder logic or control logic

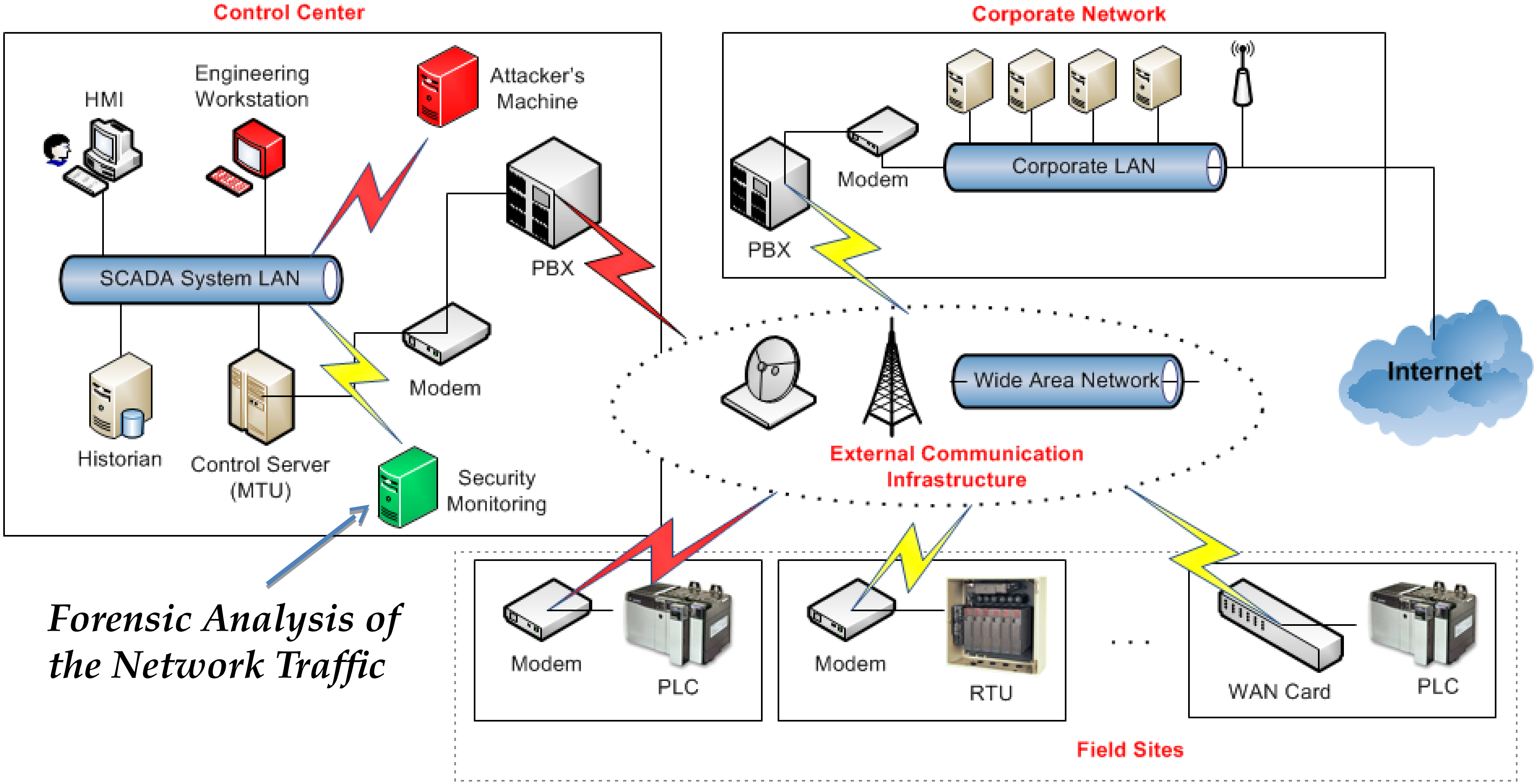
SCADA System Overview



Attacker on SCADA Network



Network Monitoring



Forensic Analysis of the Network Traffic

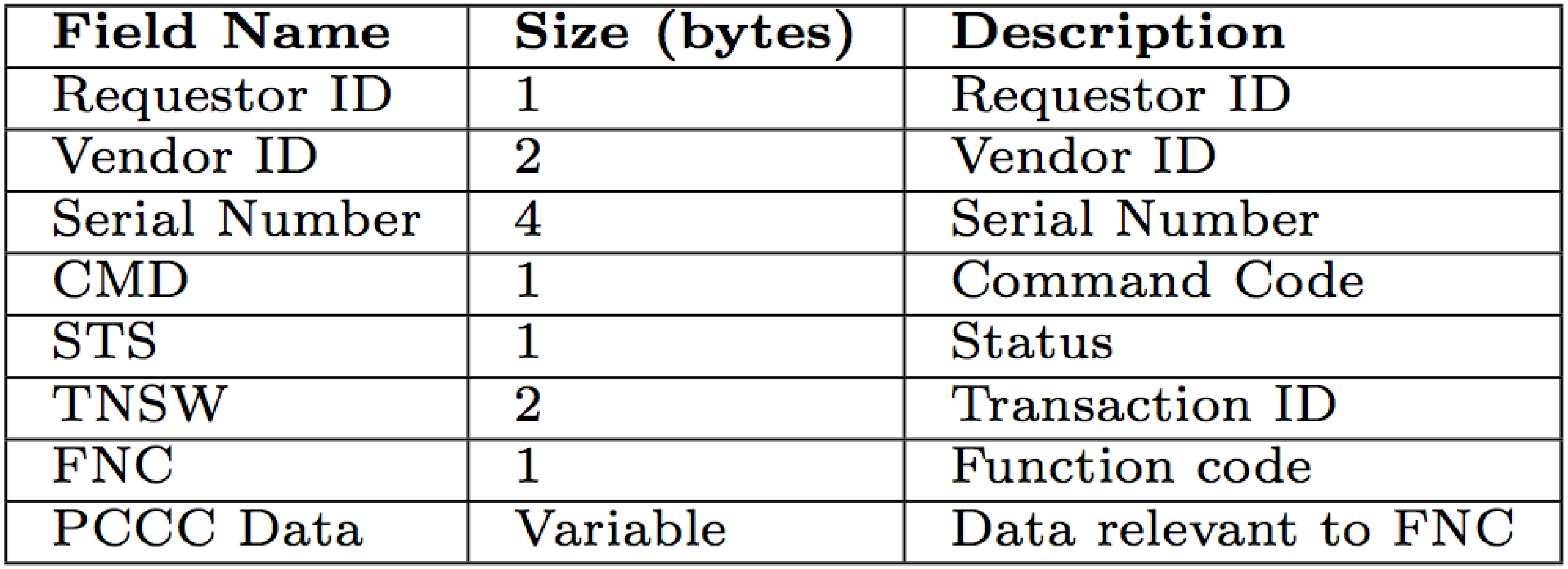
Goals

- ***Explore*** the transfer process of control logic to a PLC
    - using PCCC protocol.
- ***Identify*** digital artifacts for forensic analysis.
- ***Develop*** a protocol specific network forensic tool, Cutter

Learning the Protocol

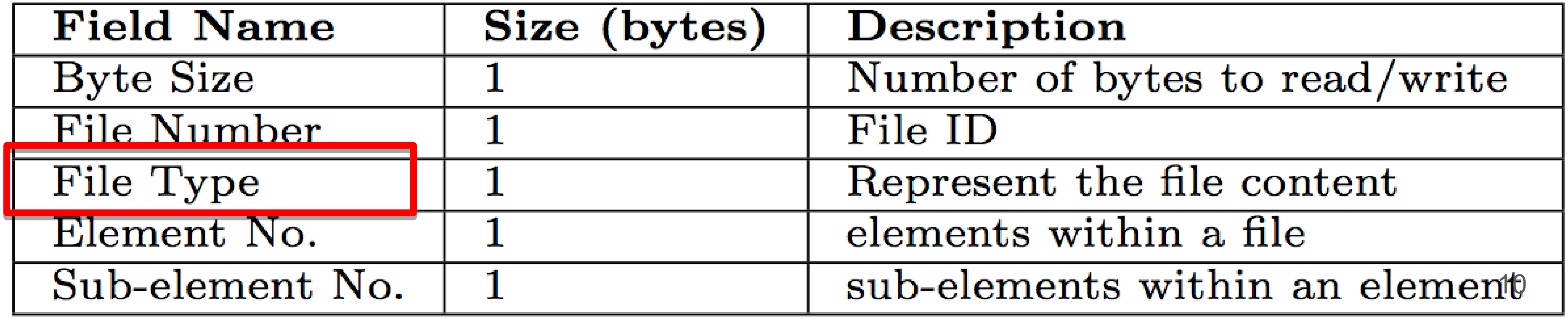
- Allen Bradley DF1 protocol and Command Set
  - http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1770-rm516_-en-p.pdf
- PLC's using the protocol ?
  - Allen Bradley Micrologix 1400 B

PCCC Message



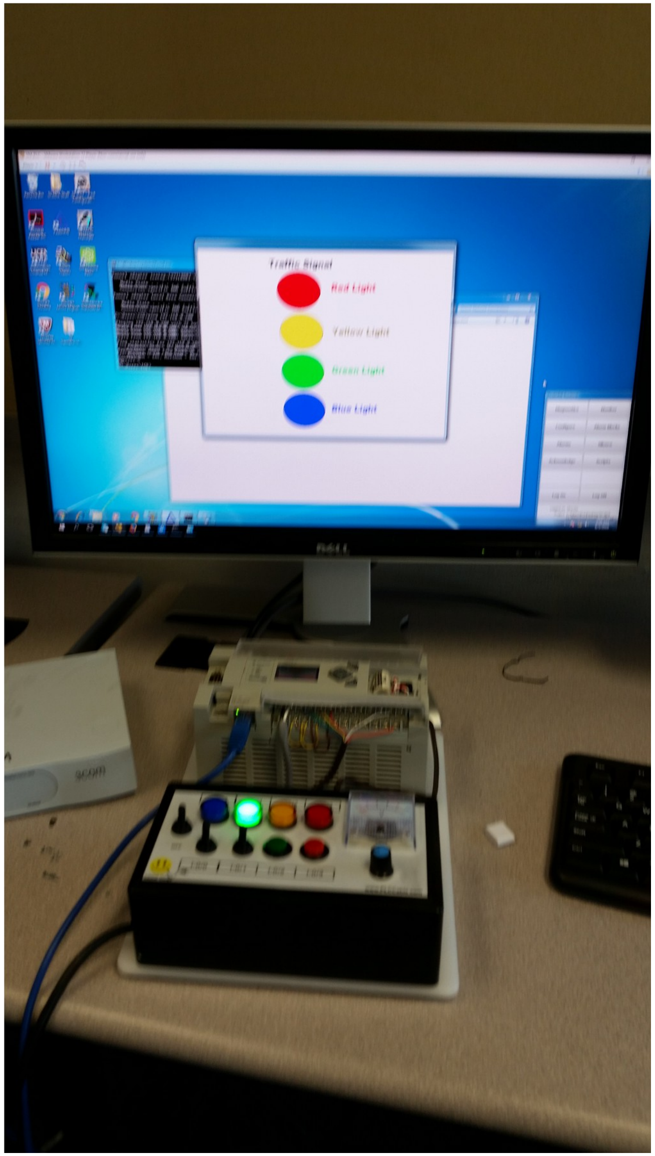
Field Name	Size (bytes)	Description
Requestor ID	1	Requestor ID
Vendor ID	2	Vendor ID
Serial Number	4	Serial Number
CMD	1	Command Code
STS	1	Status
TNSW	2	Transaction ID
FNC	1	Function code
PCCC Data	Variable	Data relevant to FNC

PCCC data field for FNC code 0xA2 and 0xAA to read/write to a PLC



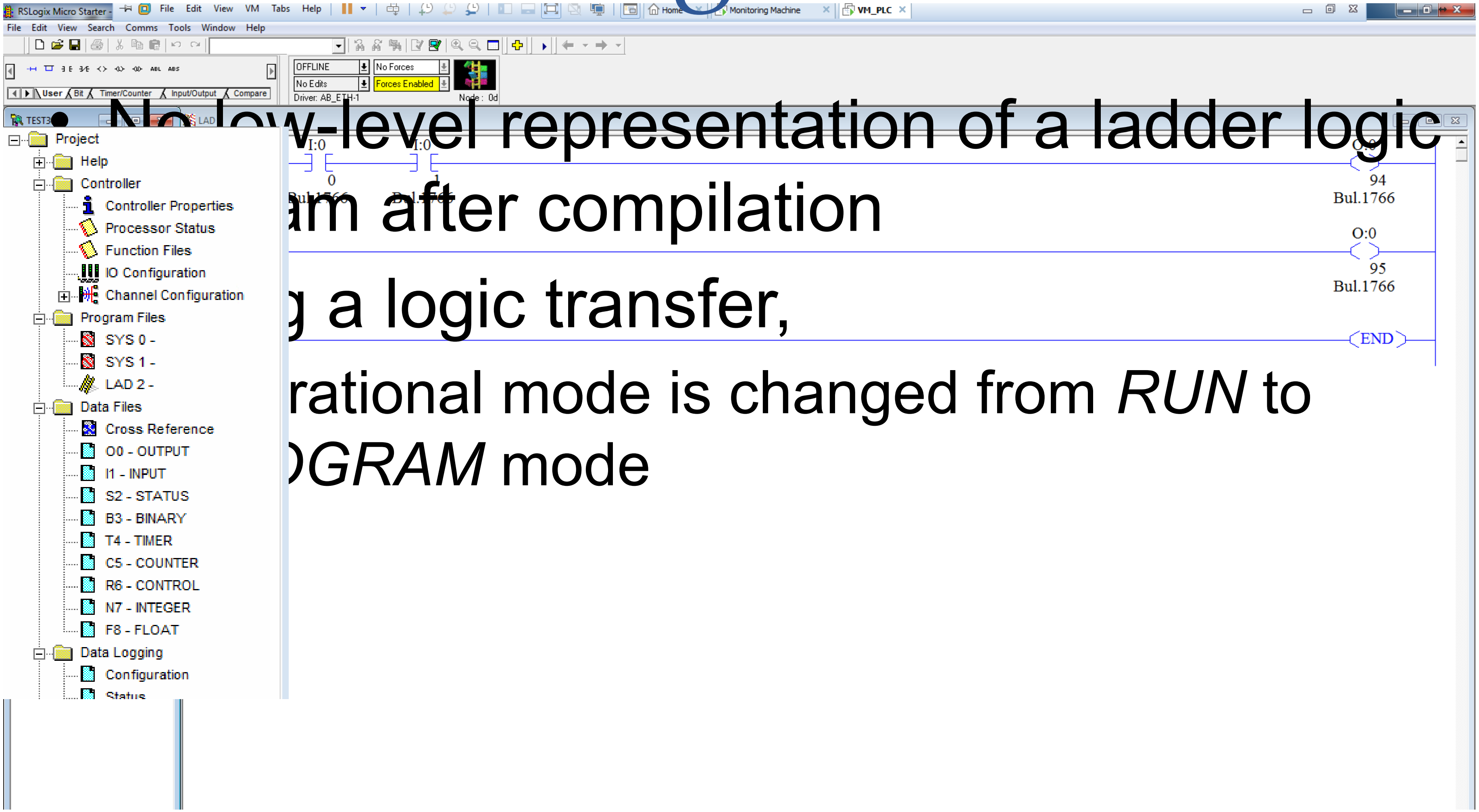
Field Name	Size (bytes)	Description
Byte Size	1	Number of bytes to read/write
File Number	1	File ID
File Type	1	Represent the file content
Element No.	1	elements within a file
Sub-element No.	1	sub-elements within an element

Experimental Setup



- PCCC - Programmable Controller Communication Commands
- PLC - **Allen-Bradley Micrologix 1400 B**
- **RSLogix 500** Programming S/W

RSLogix IDE



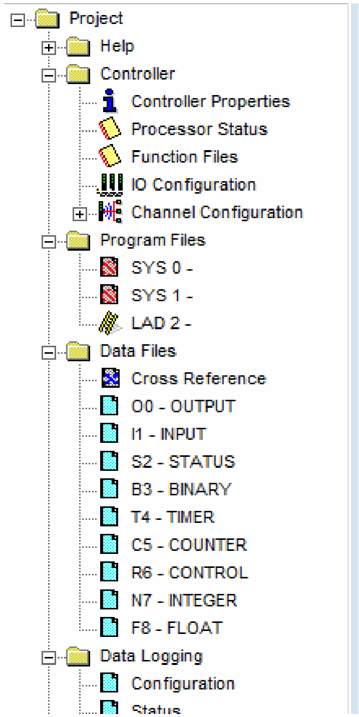
OFFLINE
No Forces
No Edits
Forces Enabled
Driver: AB_ETH-1
Node : 0d

- Project
  - Help
  - Controller
    - Controller Properties
    - Processor Status
    - Function Files
    - IO Configuration
    - Channel Configuration
  - Program Files
    - SYS 0 -
    - SYS 1 -
    - LAD 2 -
  - Data Files
    - Cross Reference
    - O0 - OUTPUT
    - I1 - INPUT
    - S2 - STATUS
    - B3 - BINARY
    - T4 - TIMER
    - C5 - COUNTER
    - R6 - CONTROL
    - N7 - INTEGER
    - F8 - FLOAT
  - Data Logging
    - Configuration
    - Status

- No low-level representation of a ladder logic am after compilation g a logic transfer, rational mode is changed from *RUN* to *OGRAM* mode

RSLogix IDE

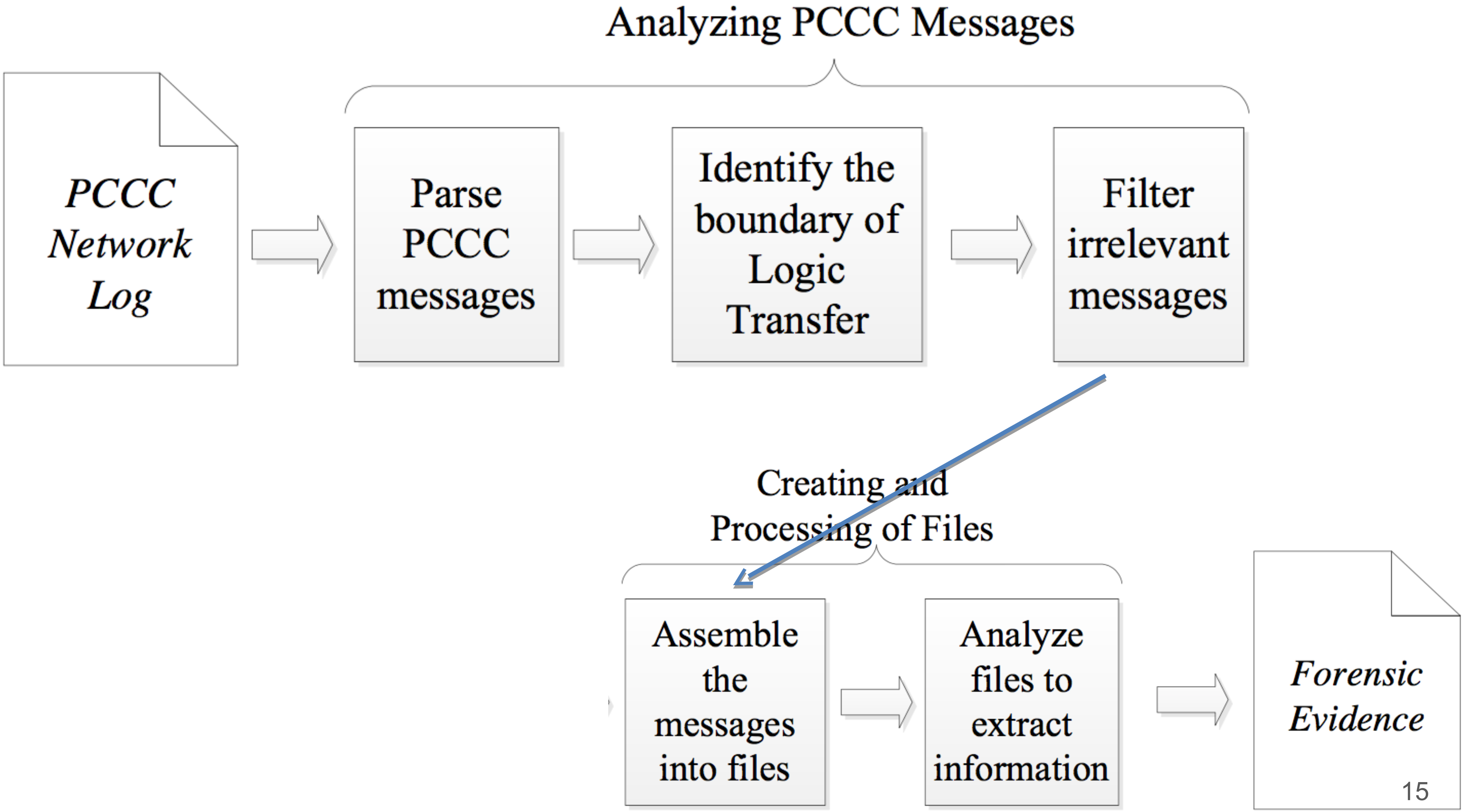
- 30 types of files are transferred to PLC
  - During ladder logic transfer
- File types of data files are known
- Other unknown types are
  - System configuration
  - Ladder logic



- Project
  - Help
  - Controller
    - Controller Properties
    - Processor Status
    - Function Files
    - IO Configuration
    - Channel Configuration
  - Program Files
    - SYS 0 -
    - SYS 1 -
    - LAD 2 -
  - Data Files
    - Cross Reference
    - O0 - OUTPUT
    - I1 - INPUT
    - S2 - STATUS
    - B3 - BINARY
    - T4 - TIMER
    - C5 - COUNTER
    - R6 - CONTROL
    - N7 - INTEGER
    - F8 - FLOAT
  - Data Logging
    - Configuration
    - Status

Implementation

Cutter tool – Modules



Analyzing PCCC Messages

PCCC Network Log

Parse PCCC messages

Identify the boundary of Logic Transfer

Filter irrelevant messages

Creating and Processing of Files

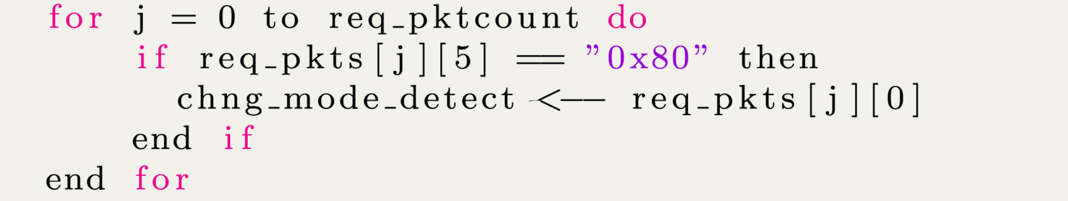
Assemble the messages into files

Analyze files to extract information

Forensic Evidence

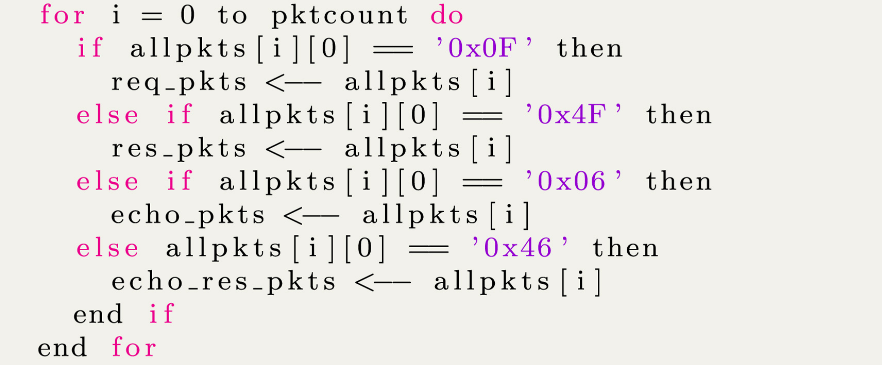
Cutter tool – Modules

- Identify the boundary of the logic transfer



```
for j = 0 to req_pktcount do
    if req_pkts[j][5] == ”0x80” then
      chng_mode_detect <-- req_pkts[j][0]
    end if
end for
```

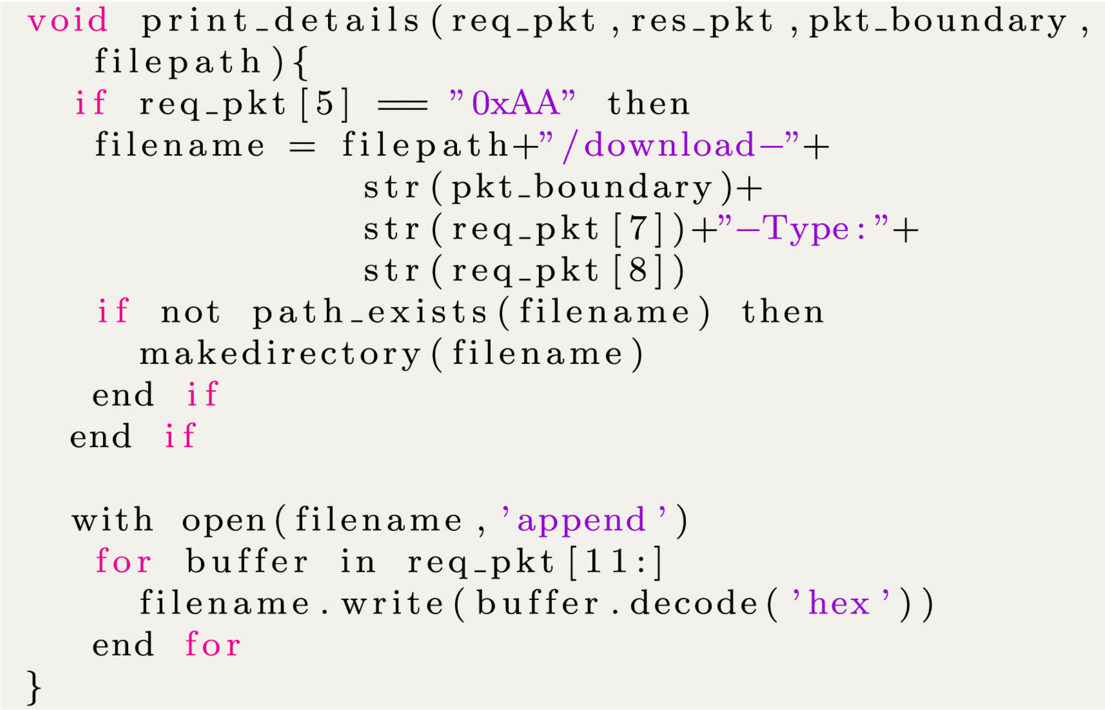
- Filter Irrelevant messages



```
for i = 0 to pktcount do
  if allpkts[i][0] == '0x0F' then
    req_pkts <-- allpkts[i]
  else if allpkts[i][0] == '0x4F' then
    res_pkts <-- allpkts[i]
  else if allpkts[i][0] == '0x06' then
    echo_pkts <-- allpkts[i]
  else allpkts[i][0] == '0x46' then
    echo_res_pkts <-- allpkts[i]
  end if
end for
```

Cutter tool – Modules

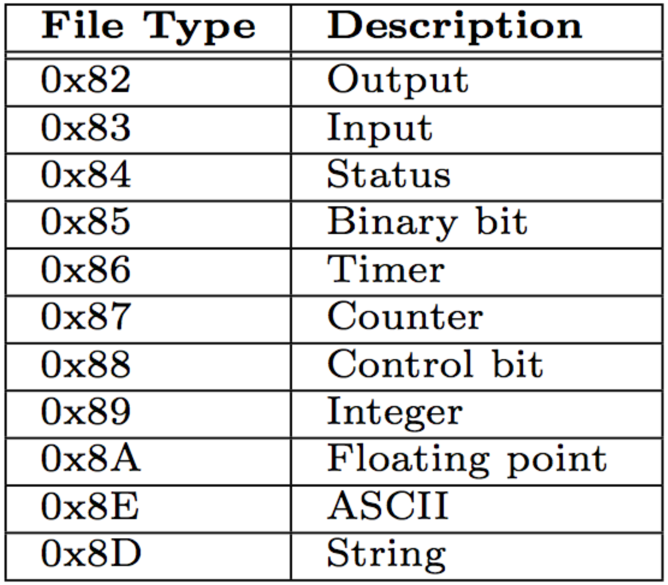
- Assemble the write messages into files
  - File number(xx) and Filetype(yy) combination is used to create a unique file name → file:xx-Type:yy



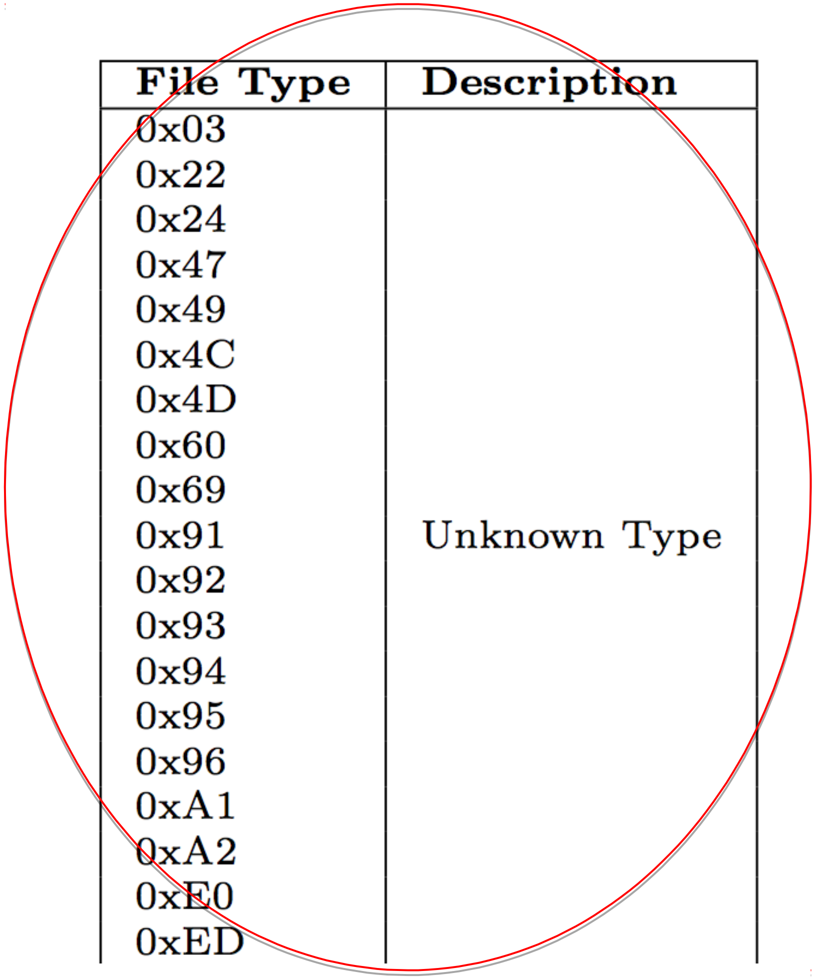
```
void print_details(req_pkt,res_pkt,pkt_boundary,
    filepath){
   if req_pkt[5] == "0xAA" then
    filename = filepath+"/download-"+
                str(pkt_boundary)+
                str(req_pkt[7])+"-Type:"+
                str(req_pkt[8])
    if not path_exists(filename) then
      makedirectory(filename)
    end if
  end if

  with open(filename,'append')
   for buffer in req_pkt[11:]
     filename.write(buffer.decode('hex'))
    end for
}
```

Unknown File types



File Type	Description
0x82	Output
0x83	Input
0x84	Status
0x85	Binary bit
0x86	Timer
0x87	Counter
0x88	Control bit
0x89	Integer
0x8A	Floating point
0x8E	ASCII
0x8D	String



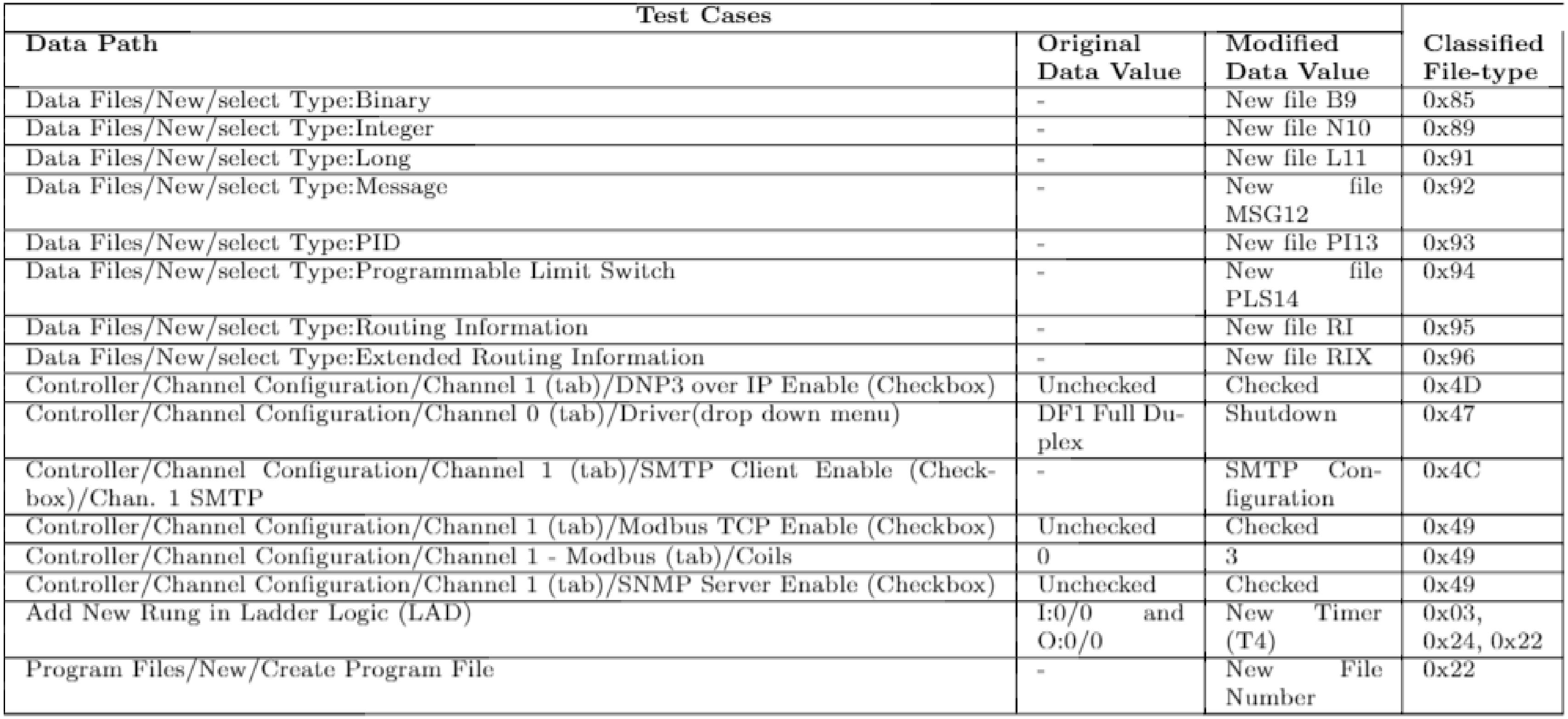
File Type	Description
0x03	Unknown Type
0x22	
0x24	
0x47	
0x49	
0x4C	
0x4D	
0x60	
0x69	
0x91	
0x92	
0x93	
0x94	
0x95	
0x96	
0xA1	
0xA2	
0xE0	
0xED	

Unknown file type Identification

- A differential Analysis approach
- Different test cases are created
- Compared incrementally with each other
  - $TestCase_{n-1} - TestCase_n$

Unknown file type Identification

- Test cases



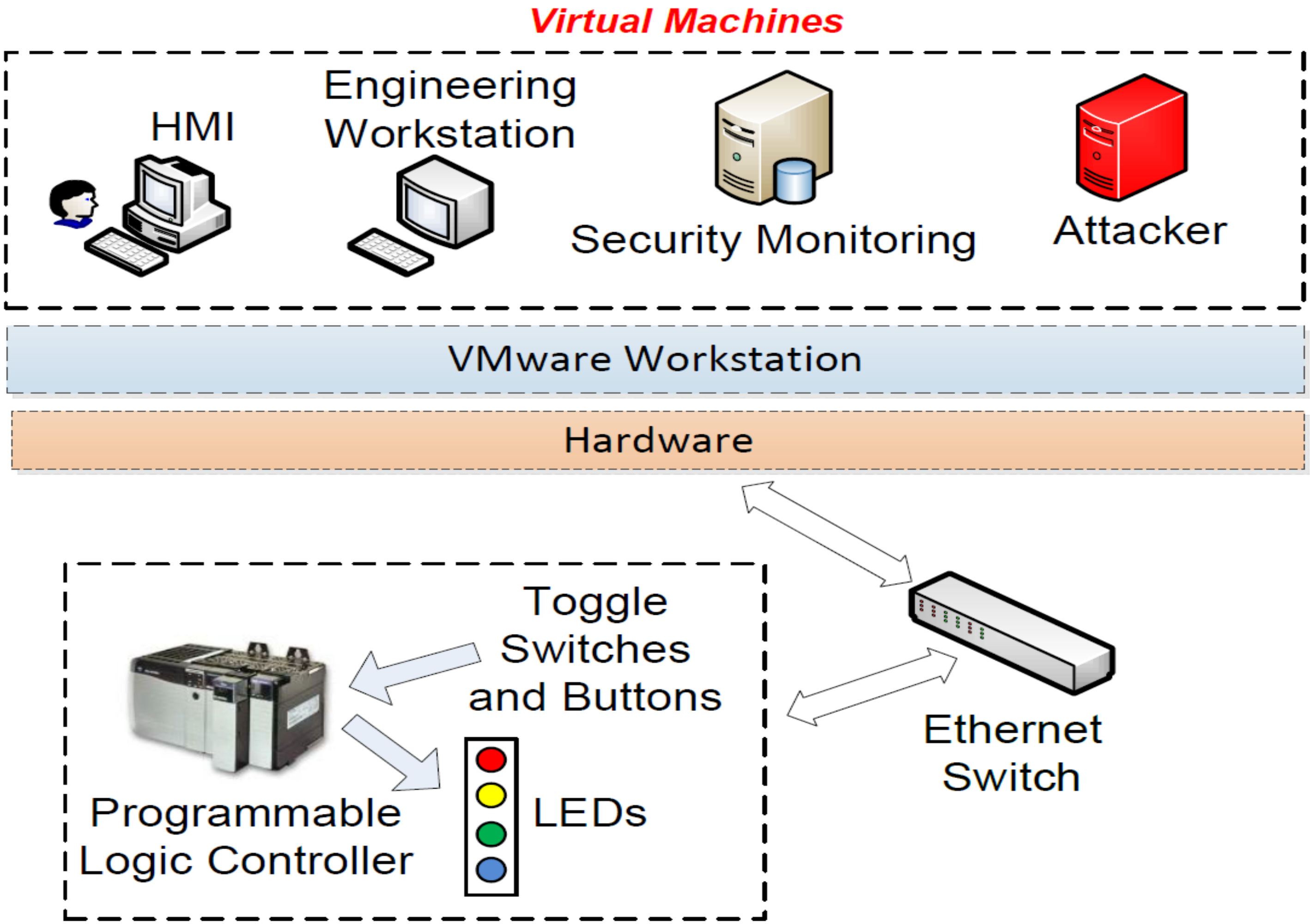
Test Cases			
Data Path	Original Data Value	Modified Data Value	Classified File-type
Data Files/New/select Type:Binary	-	New file B9	0x85
Data Files/New/select Type:Integer	-	New file N10	0x89
Data Files/New/select Type:Long	-	New file L11	0x91
Data Files/New/select Type:Message	-	New file MSG12	0x92
Data Files/New/select Type:PID	-	New file PI13	0x93
Data Files/New/select Type:Programmable Limit Switch	-	New file PLS14	0x94
Data Files/New/select Type:Routing Information	-	New file RI	0x95
Data Files/New/select Type:Extended Routing Information	-	New file RIX	0x96
Controller/Channel Configuration/Channel 1 (tab)/DNP3 over IP Enable (Checkbox)	Unchecked	Checked	0x4D
Controller/Channel Configuration/Channel 0 (tab)/Driver(drop down menu)	DF1 Full Duplex	Shutdown	0x47
Controller/Channel Configuration/Channel 1 (tab)/SMTP Client Enable (Checkbox)/Chan. 1 SMTP	-	SMTP Configuration	0x4C
Controller/Channel Configuration/Channel 1 (tab)/Modbus TCP Enable (Checkbox)	Unchecked	Checked	0x49
Controller/Channel Configuration/Channel 1 - Modbus (tab)/Coils	0	3	0x49
Controller/Channel Configuration/Channel 1 (tab)/SNMP Server Enable (Checkbox)	Unchecked	Checked	0x49
Add New Rung in Ladder Logic (LAD)	I:0/0 and O:0/0	New Timer (T4)	0x03, 0x24, 0x22
Program Files/New/Create Program File	-	New File Number	0x22

Classified File Types

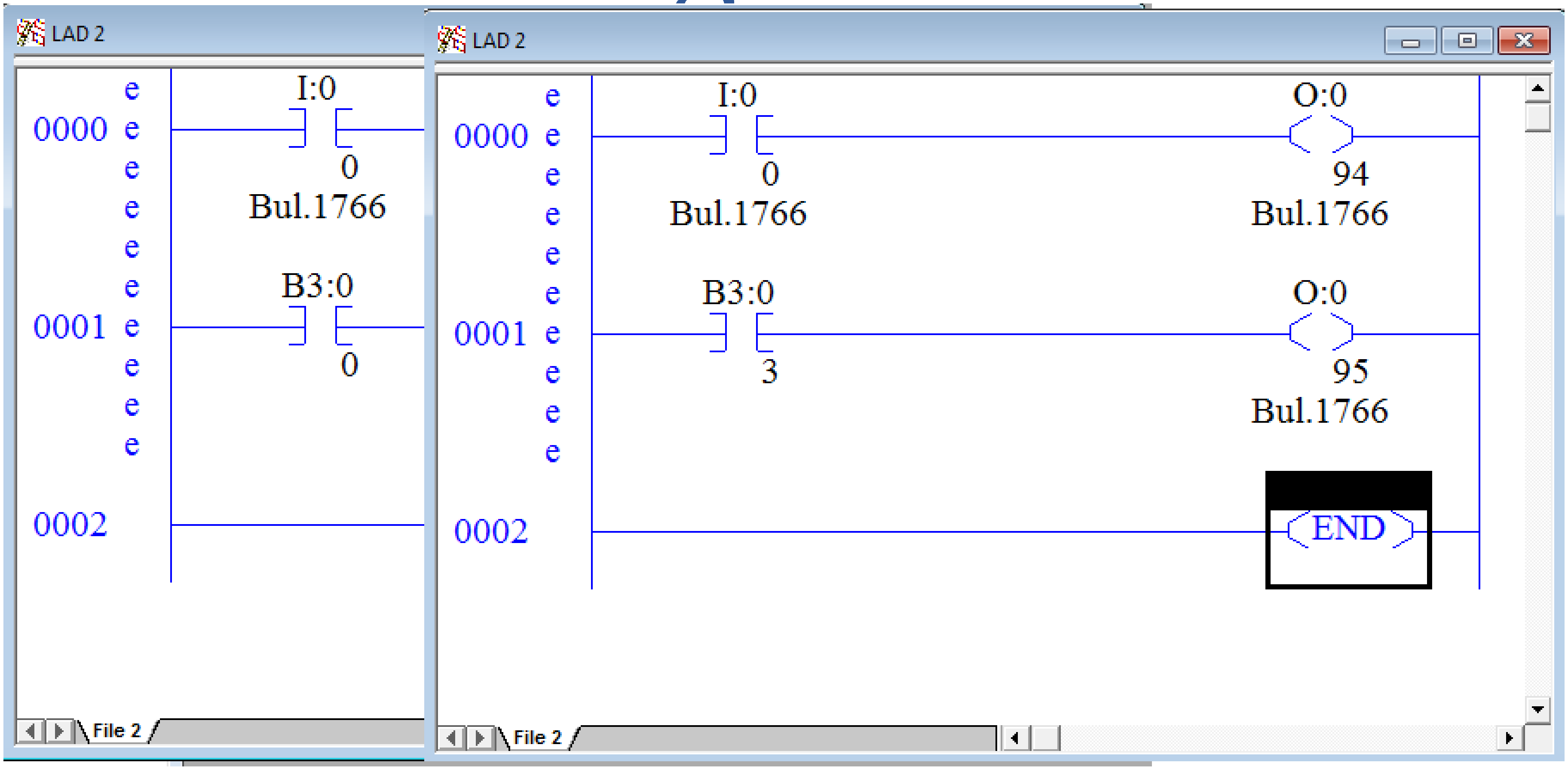
File Type	Classification (Based on Content)
0x22	Ladder Logic – Control Logic Program
0x03	Main Configuration file
0x47	DF1 (Channel 0) Configuration
0x49	Ethernet Configuration
0x4D	DNP3 Configuration
0x4C	SMTP Configuration
0x92	Message
0x93	PID
0x94	Programmable Limit Switch
0x95	Routing Information
0x96	Extended Routing Information

Evaluation

Experimental Settings

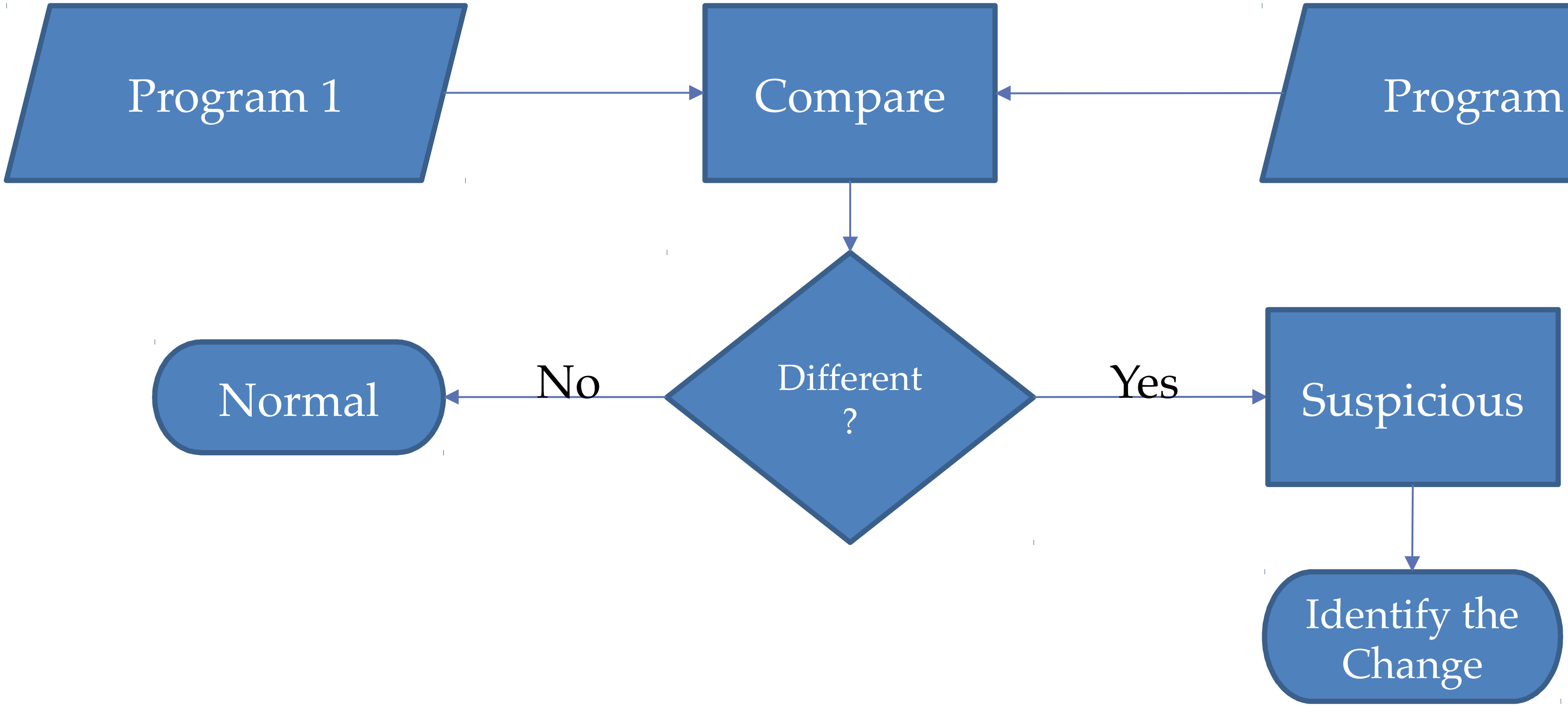


Compare Two Ladder Programs

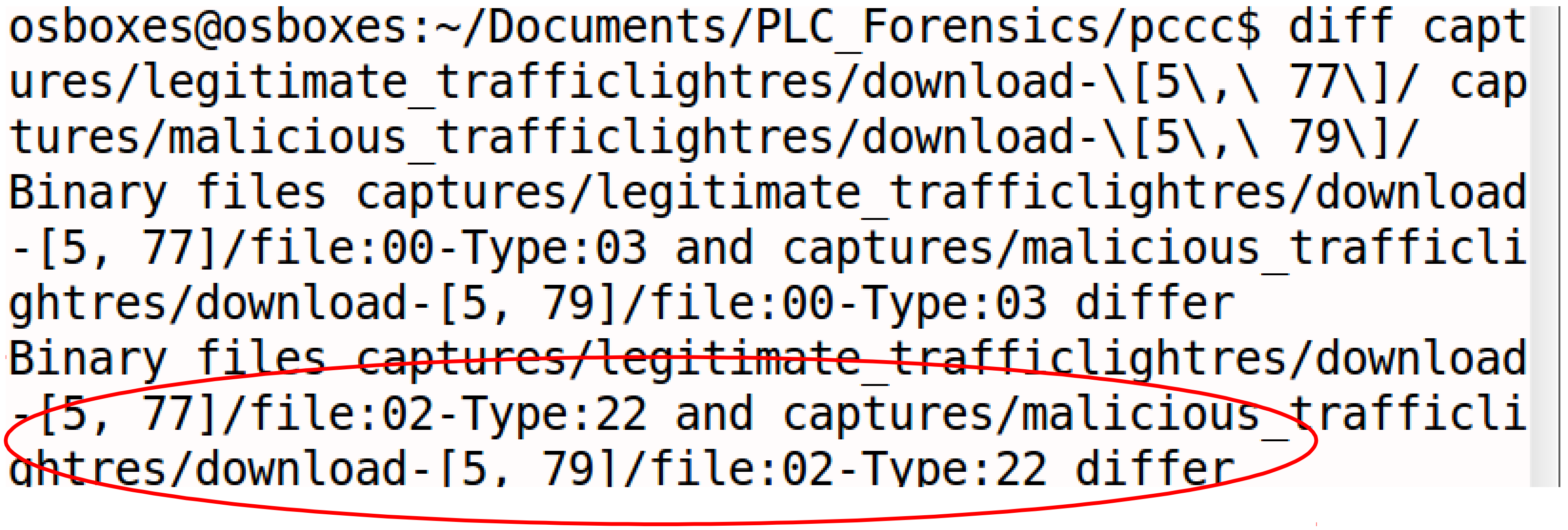


Compare Two Ladder Programs

- Program 1: *Original program in PLC*
- Program 2: *Found in a network traffic log*

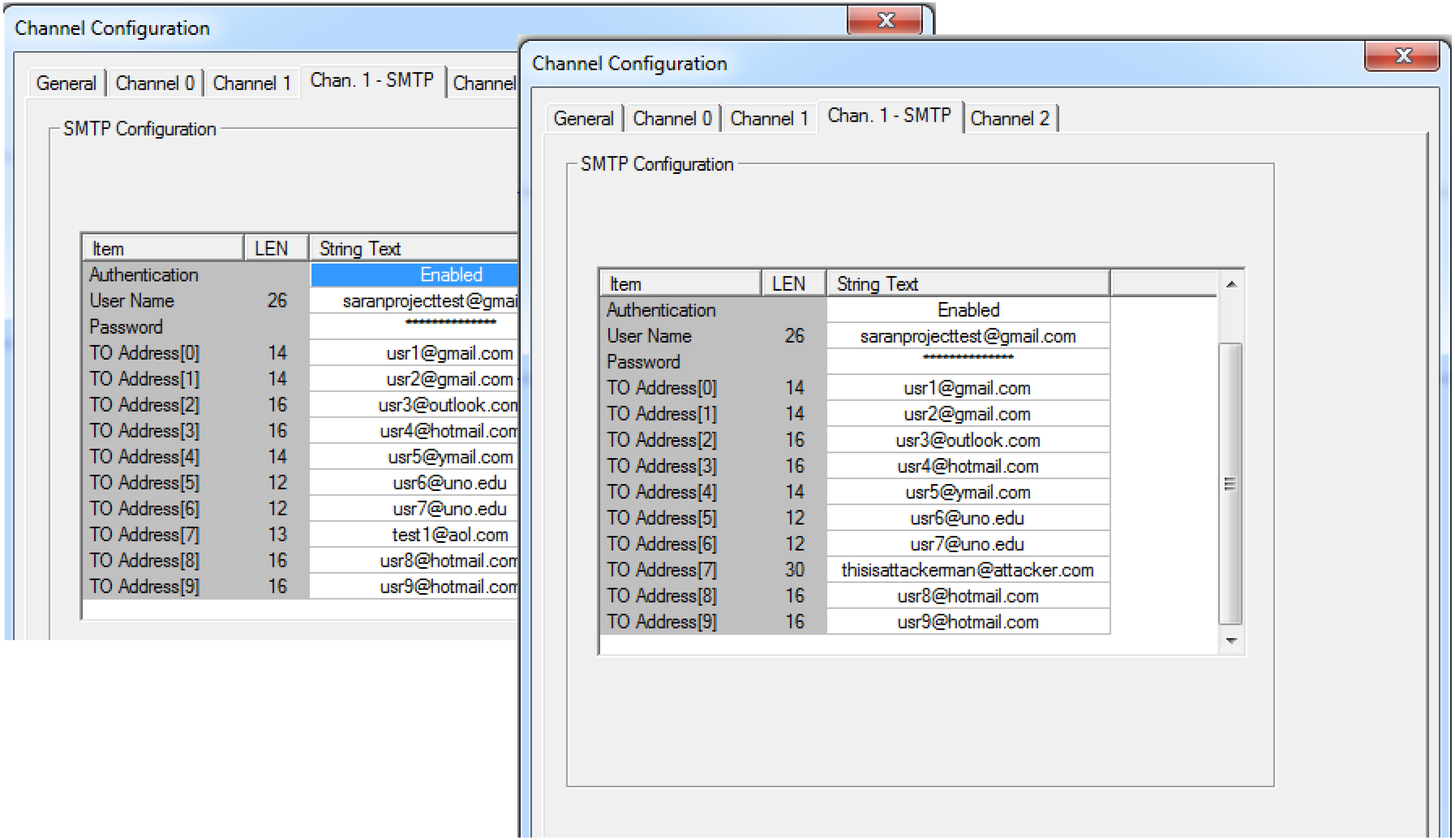


Compare Two Ladder Programs



```
osboxes@osboxes:~/Documents/PLC_Forensics/pccc$ diff capt
ures/legitimate_trafficlightres/download-\[5\,\ 77\]/ cap
tures/malicious_trafficlightres/download-\[5\,\ 79\]/
Binary files captures/legitimate_trafficlightres/download
-[5, 77]/file:00-Type:03 and captures/malicious_traffici
ghtres/download-[5, 79]/file:00-Type:03 differ
Binary files captures/legitimate_trafficlightres/download
-[5, 77]/file:02-Type:22 and captures/malicious_traffici
ghtres/download-[5, 79]/file:02-Type:22 differ
```

Compare Two SMTP Files

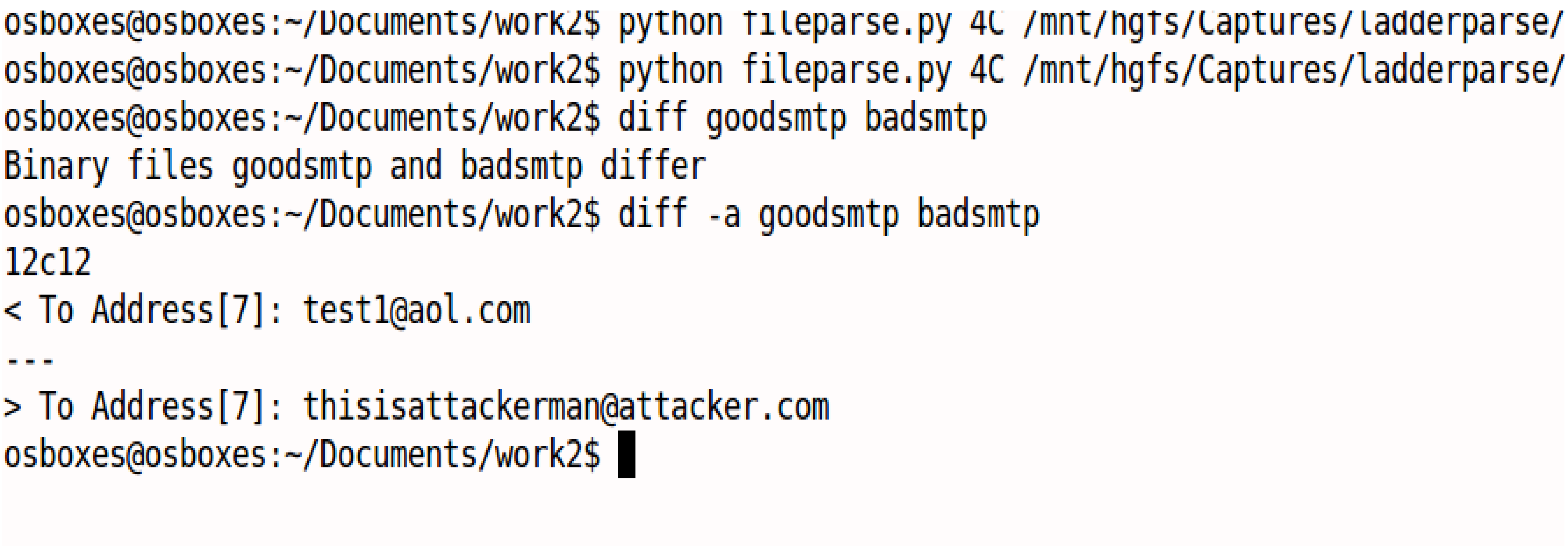


Item	LEN	String Text
Authentication		Enabled
User Name	26	saranprojecttest@gmai
Password		***************
TO Address[0]	14	usr1@gmail.com
TO Address[1]	14	usr2@gmail.com
TO Address[2]	16	usr3@outlook.con
TO Address[3]	16	usr4@hotmail.con
TO Address[4]	14	usr5@ymail.com
TO Address[5]	12	usr6@uno.edu
TO Address[6]	12	usr7@uno.edu
TO Address[7]	13	test1@aol.com
TO Address[8]	16	usr8@hotmail.con
TO Address[9]	16	usr9@hotmail.con

Item	LEN	String Text
Authentication		Enabled
User Name	26	saranprojecttest@gmail.com
Password		***************
TO Address[0]	14	usr1@gmail.com
TO Address[1]	14	usr2@gmail.com
TO Address[2]	16	usr3@outlook.com
TO Address[3]	16	usr4@hotmail.com
TO Address[4]	14	usr5@ymail.com
TO Address[5]	12	usr6@uno.edu
TO Address[6]	12	usr7@uno.edu
TO Address[7]	30	thisisattackerman@attacker.com
TO Address[8]	16	usr8@hotmail.com
TO Address[9]	16	usr9@hotmail.com

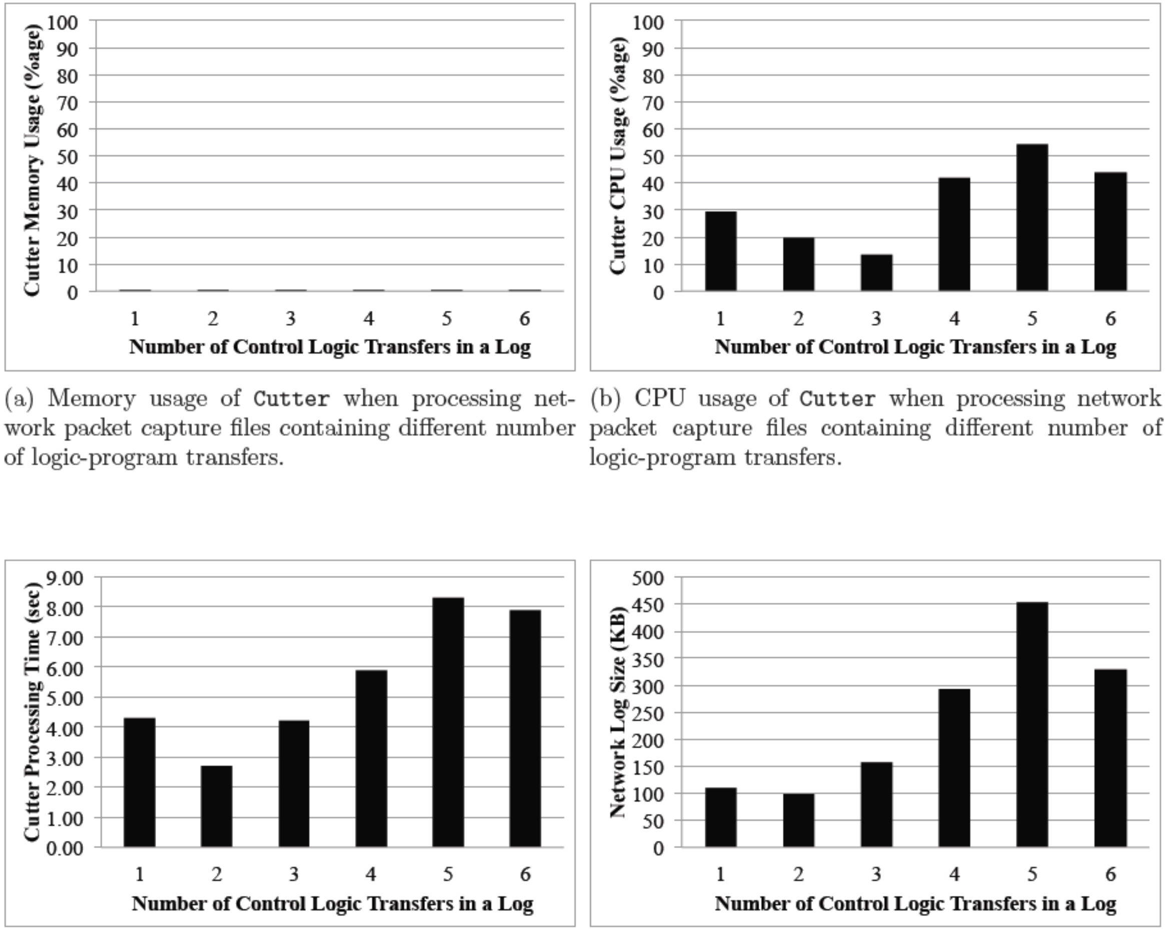
Compare Two SMTP Files

- SMTP Config 1: *Original configuration in PLC*
- SMTP Config 2: *Found in a network traffic log*



```
osboxes@osboxes:~/Documents/work2$ python fileparse.py 4C /mnt/hgfs/Captures/ladderparse/
osboxes@osboxes:~/Documents/work2$ python fileparse.py 4C /mnt/hgfs/Captures/ladderparse/
osboxes@osboxes:~/Documents/work2$ diff goodsmtp badsmtp
Binary files goodsmtp and badsmtp differ
osboxes@osboxes:~/Documents/work2$ diff -a goodsmtp badsmtp
12c12
< To Address[7]: test1@aol.com
---
> To Address[7]: thisisattackerman@attacker.com
osboxes@osboxes:~/Documents/work2$ █
```

Performance Evaluation



(a) Memory usage of `Cutter` when processing network packet capture files containing different number of logic-program transfers.

(b) CPU usage of `Cutter` when processing network packet capture files containing different number of logic-program transfers.

(c) Processing time of packet network capture containing different number of logic-program transfers.

(d) Size of packet network capture files containing different number of logic-program transfers.

Figure 4.5: Performance Evaluation of `Cutter`

Conclusion & Future Work

- Framework developed for SCADA forensic analysis
- Future Works
  - Parsing the assembled binary file into human Readable format
    - Includes disassembling the Ladder logic file 0x22
  - Framework could be developed for universal applicability
    - Modbus, DNP3 etc.,

Questions

- Tool will be available at → https://gitlab.cs.uno.edu/ssenthiv/PLC_Forensics.git

Thank You