The spear to break the security wall of S7CommPlus

CHENG LEI， NSFOCUS

Related Work

- Dillon Beresford. Exploiting Siemens Simatic S7 PLCs. Black Hat 2011 USA.

  S7Comm protocol
- Ralf Spenneberg et. al.

  PLC-Blaster: A Worm Living Solely in the PLC. Black Hat 2016 USA

  Early S7CommPlus protocol
- This talk mainly focus on the current encrypted S7CommPlus protocol

What is PLC

Programmable Logic Controllers (PLC) is responsible for process control in industrial control system. A PLC contains a Central Processing Unit (CPU), some digital/analog inputs and outputs modules, communication module and some process modules like PID.

Siemens PLCs

S7-300

- S7-200,S7-300,S7-400 using the S7Comm protocol

S7-1200

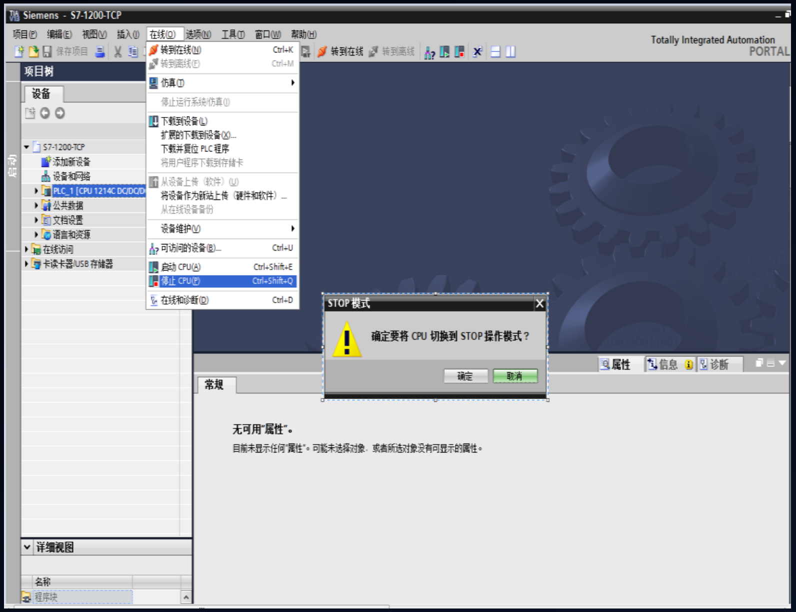
- S7-1200v3.0 using the early S7CommPlus protocol

S7-1500

- S7-1200v4.0, S7-1500 using the current encrypted S7CommPlus protocol

TIA Portal

TIA Portal is the configuration and programming software for Siemens PLCs.



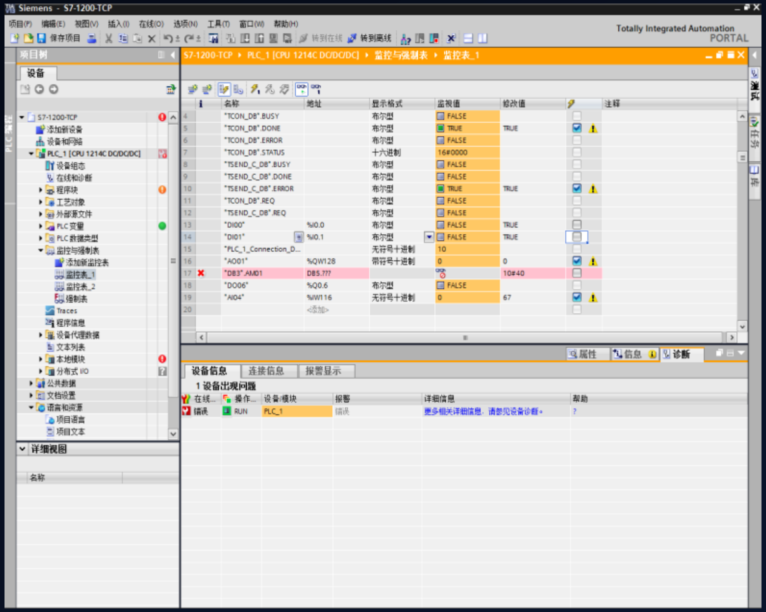
STOP 模式

确定要将 CPU 切换到 STOP 操作模式？

确定 取消

无可用"属性"。

目前未显示任何"属性"。可能未选择对象，或者所选对象没有可显示的属性。



S7-1200-TCP ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ 监控与强制表 ▸ 监控表_1

	i	名称	地址	显示格式	监视值	修改值	⚡		注释
4		"TCON_DB".BUSY		布尔型	FALSE		□		
5		"TCON_DB".DONE		布尔型	TRUE	TRUE	☑	⚠	
6		"TCON_DB".ERROR		布尔型	FALSE		□		
7		"TCON_DB".STATUS		十六进制	16#0000		□		
8		"TSEND_C_DB".BUSY		布尔型	FALSE		□		
9		"TSEND_C_DB".DONE		布尔型	FALSE		□		
10		"TSEND_C_DB".ERROR		布尔型	TRUE	TRUE	☑	⚠	
11		"TCON_DB".REQ		布尔型	FALSE		□		
12		"TSEND_C_DB".REQ		布尔型	FALSE		□		
13		"DI00"	%I0.0	布尔型	FALSE	TRUE	□		
14		"DI01"	%I0.1	布尔型	FALSE	TRUE	□		
15		"PLC_1_Connection_D...		无符号十进制	10		□		
16		"AO01"	%QW128	带符号十进制	0	0	☑	⚠	
17	✗	"DB3".AM01	DB5.???			10#40	□		
18		"DO06"	%Q0.6	布尔型	FALSE		□		
19		"AI04"	%IW116	无符号十进制	0	67	☑	⚠	
20			<添加>				□		

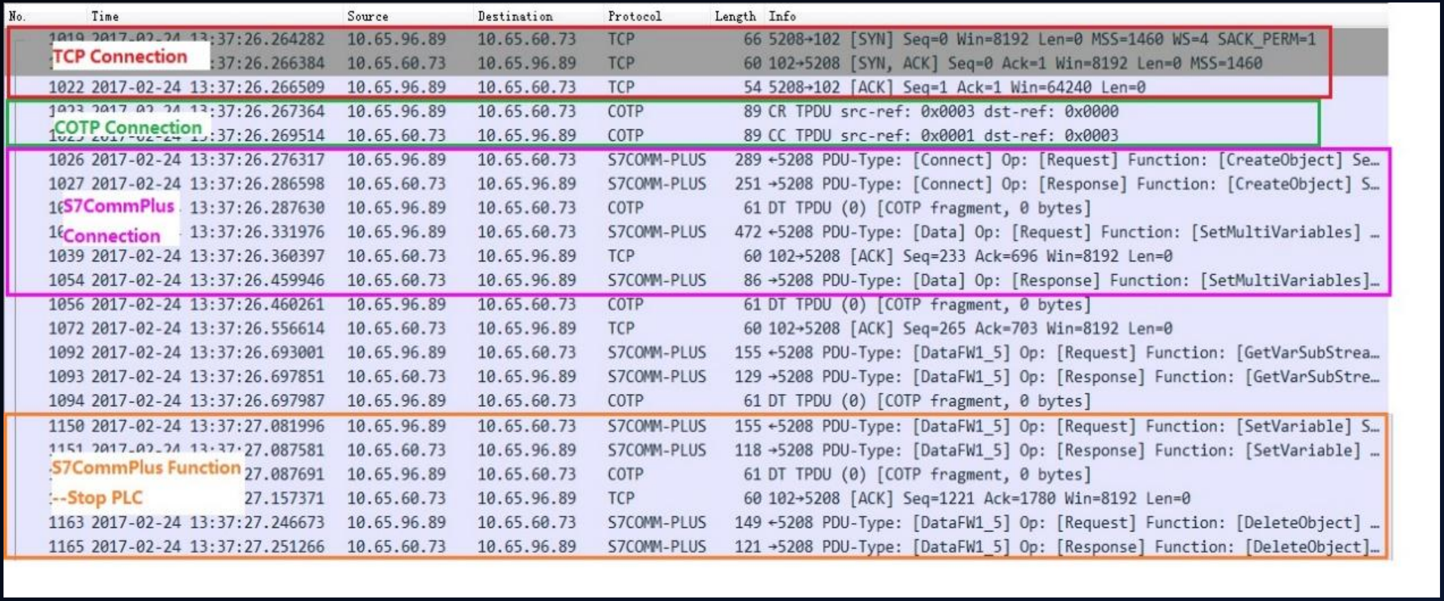
设备信息 连接信息 报警显示

1 设备出现问题

在线...	操作...	设备/模块	报警	详细信息	帮助
错误	RUN	PLC_1	错误	更多相关详细信息，请参见设备诊断。	?

Replay Attack

- Replay attacks have been widely used in PLC attacks.
- Get the communication sequence packets with the help of Wireshark



TCP Connection

COTP Connection

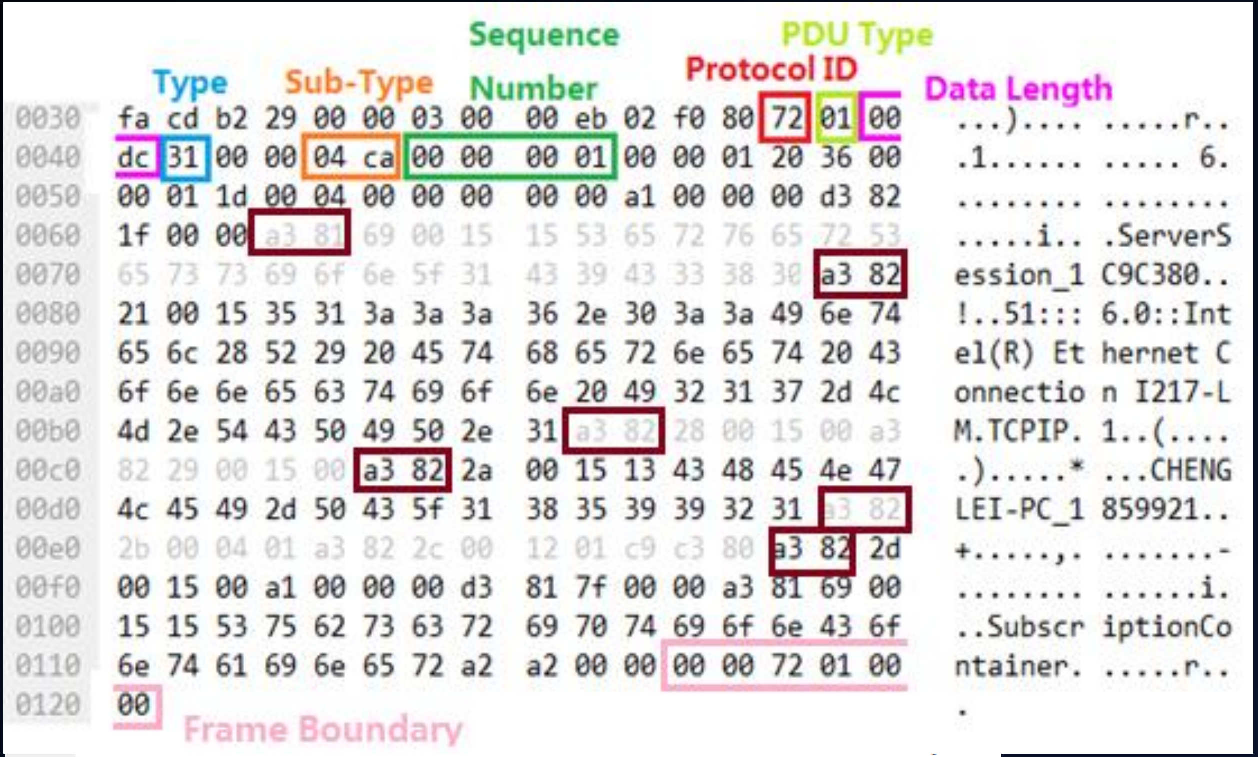
S7CommPlus Connection

S7CommPlus Function --Stop PLC

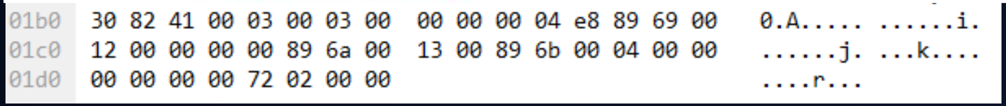
No.	Time	Source	Destination	Protocol	Length	Info
1019	2017-02-24 13:37:26.264282	10.65.96.89	10.65.60.73	TCP	66	5208→102 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
[illegible]	:37:26.266384	10.65.60.73	10.65.96.89	TCP	60	102→5208 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
1022	2017-02-24 13:37:26.266509	10.65.96.89	10.65.60.73	TCP	54	5208→102 [ACK] Seq=1 Ack=1 Win=64240 Len=0
1023	2017-02-24 13:37:26.267364	10.65.96.89	10.65.60.73	COTP	89	CR TPDU src-ref: 0x0003 dst-ref: 0x0000
1025	2017-02-24 13:37:26.269514	10.65.60.73	10.65.96.89	COTP	89	CC TPDU src-ref: 0x0001 dst-ref: 0x0003
1026	2017-02-24 13:37:26.276317	10.65.96.89	10.65.60.73	S7COMM-PLUS	289	←5208 PDU-Type: [Connect] Op: [Request] Function: [CreateObject] Se…
1027	2017-02-24 13:37:26.286598	10.65.60.73	10.65.96.89	S7COMM-PLUS	251	→5208 PDU-Type: [Connect] Op: [Response] Function: [CreateObject] S…
1[illegible]	13:37:26.287630	10.65.96.89	10.65.60.73	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
1[illegible]	13:37:26.331976	10.65.96.89	10.65.60.73	S7COMM-PLUS	472	←5208 PDU-Type: [Data] Op: [Request] Function: [SetMultiVariables] …
1039	2017-02-24 13:37:26.360397	10.65.60.73	10.65.96.89	TCP	60	102→5208 [ACK] Seq=233 Ack=696 Win=8192 Len=0
1054	2017-02-24 13:37:26.459946	10.65.60.73	10.65.96.89	S7COMM-PLUS	86	→5208 PDU-Type: [Data] Op: [Response] Function: [SetMultiVariables]…
1056	2017-02-24 13:37:26.460261	10.65.96.89	10.65.60.73	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
1072	2017-02-24 13:37:26.556614	10.65.60.73	10.65.96.89	TCP	60	102→5208 [ACK] Seq=265 Ack=703 Win=8192 Len=0
1092	2017-02-24 13:37:26.693001	10.65.96.89	10.65.60.73	S7COMM-PLUS	155	←5208 PDU-Type: [DataFW1_5] Op: [Request] Function: [GetVarSubStrea…
1093	2017-02-24 13:37:26.697851	10.65.60.73	10.65.96.89	S7COMM-PLUS	129	→5208 PDU-Type: [DataFW1_5] Op: [Response] Function: [GetVarSubStre…
1094	2017-02-24 13:37:26.697987	10.65.96.89	10.65.60.73	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
1150	2017-02-24 13:37:27.081996	10.65.96.89	10.65.60.73	S7COMM-PLUS	155	←5208 PDU-Type: [DataFW1_5] Op: [Request] Function: [SetVariable] S…
1151	2017-02-24 13:37:27.087581	10.65.60.73	10.65.96.89	S7COMM-PLUS	118	→5208 PDU-Type: [DataFW1_5] Op: [Response] Function: [SetVariable] …
[illegible]	27.087691	10.65.96.89	10.65.60.73	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
[illegible]	27.157371	10.65.60.73	10.65.96.89	TCP	60	102→5208 [ACK] Seq=1221 Ack=1780 Win=8192 Len=0
1163	2017-02-24 13:37:27.246673	10.65.96.89	10.65.60.73	S7COMM-PLUS	149	←5208 PDU-Type: [DataFW1_5] Op: [Request] Function: [DeleteObject] …
1165	2017-02-24 13:37:27.251266	10.65.60.73	10.65.96.89	S7COMM-PLUS	121	→5208 PDU-Type: [DataFW1_5] Op: [Response] Function: [DeleteObject]…

S7CommPlus Protocol

- The current S7CommPlus protocol including the S7CommPlus Connection packets and S7CommPlus Function packets has a similar structure.
- 2. First Connection Setup Response
  4. Function Packet-Set Request



```
0030  fa cd b2 29 00 00 03 00  00 eb 02 f0 80 72 01 00   ...).... .....r..
0040  dc 31 00 00 04 ca 00 00  00 01 00 00 01 20 36 00   .1...... ..... 6.
0050  00 01 1d 00 04 00 00 00  00 00 a1 00 00 00 d3 82   ........ ........
0060  1f 00 00 a3 81 69 00 15  15 53 65 72 76 65 72 53   .....i.. .ServerS
0070  65 73 73 69 6f 6e 5f 31  43 39 43 33 38 30 a3 82   ession_1 C9C380..
0080  21 00 15 35 31 3a 3a 3a  36 2e 30 3a 3a 49 6e 74   !..51::: 6.0::Int
0090  65 6c 28 52 29 20 45 74  68 65 72 6e 65 74 20 43   el(R) Et hernet C
00a0  6f 6e 6e 65 63 74 69 6f  6e 20 49 32 31 37 2d 4c   onnectio n I217-L
00b0  4d 2e 54 43 50 49 50 2e  31 a3 82 28 00 15 00 a3   M.TCPIP. 1..(....
00c0  82 29 00 15 00 a3 82 2a  00 15 13 43 48 45 4e 47   .).....* ...CHENG
00d0  4c 45 49 2d 50 43 5f 31  38 35 39 39 32 31 a3 82   LEI-PC_1 859921..
00e0  2b 00 04 01 a3 82 2c 00  12 01 c9 c3 80 a3 82 2d   +.....,. .......-
00f0  00 15 00 a1 00 00 00 d3  81 7f 00 00 a3 81 69 00   ........ ......i.
0100  15 15 53 75 62 73 63 72  69 70 74 69 6f 6e 43 6f   ..Subscr iptionCo
0110  6e 74 61 69 6e 65 72 a2  a2 00 00 00 00 72 01 00   ntainer. .....r..
0120  00                                                 .
```

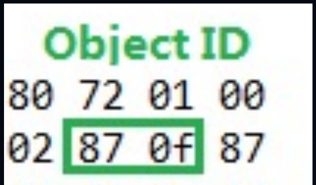


```
01b0  30 82 41 00 03 00 03 00  00 00 00 04 e8 89 69 00   0.A..... ......i.
01c0  12 00 00 00 00 89 6a 00  13 00 89 6b 00 04 00 00   ......j. ...k....
01d0  00 00 00 00 72 02 00 00                            ....r...
```

S7CommPlus Protocol

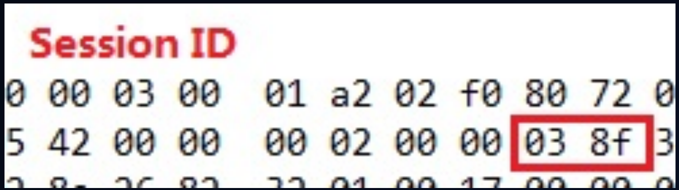
- Session ID :

Session ID = Object ID+0x80



Object ID

```
80 72 01 00
02 87 0f 87
```



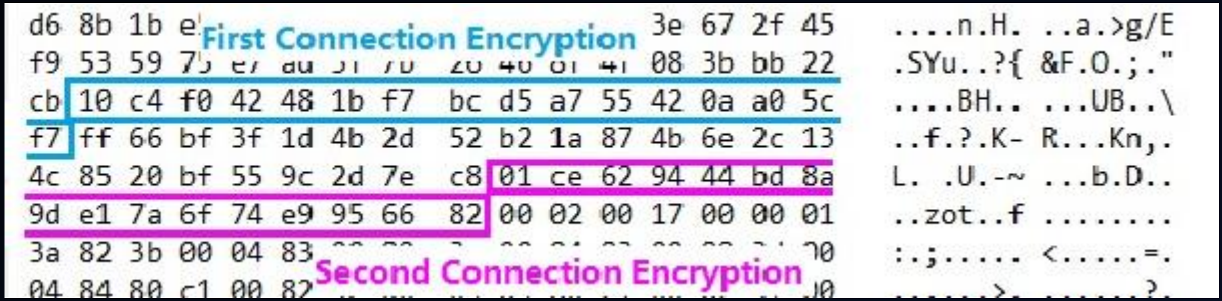
Session ID

```
0 00 03 00  01 a2 02 f0 80 72 0
5 42 00 00  00 02 00 00 03 8f 3
```

S7CommPlus Protocol

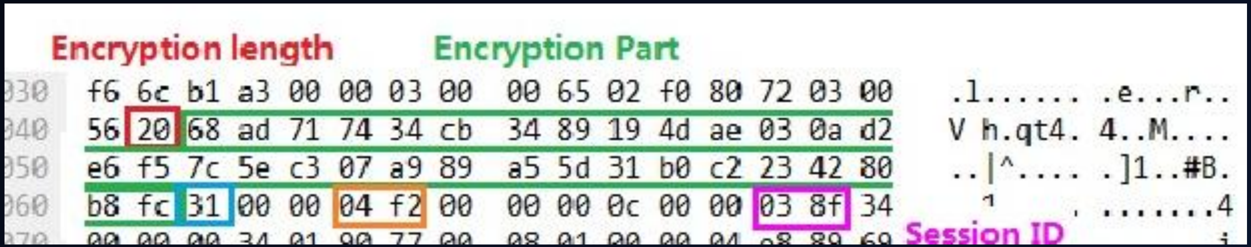
- Encryption Part :

1. The second connection packet has two encryptions



```
d6 8b 1b e                          3e 67 2f 45    ....n.H. ..a.>g/E
f9 53 59 7                          08 3b bb 22    .SYu..?{ &F.O.;."
cb 10 c4 f0 42 48 1b f7  bc d5 a7 55 42 0a a0 5c    ....BH.. ...UB..\
f7 ff 66 bf 3f 1d 4b 2d  52 b2 1a 87 4b 6e 2c 13    ..f.?.K- R...Kn,.
4c 85 20 bf 55 9c 2d 7e  c8 01 ce 62 94 44 bd 8a    L. .U.-~ ...b.D..
9d e1 7a 6f 74 e9 95 66  82 00 02 00 17 00 00 01    ..zot..f ........
3a 82 3b 00 04 83                             0    :.;..... <.....=.
04 84 80 c1 00 82                                  ......>. ......?.
```

2. The function packet has one encryption (Integrity Part)



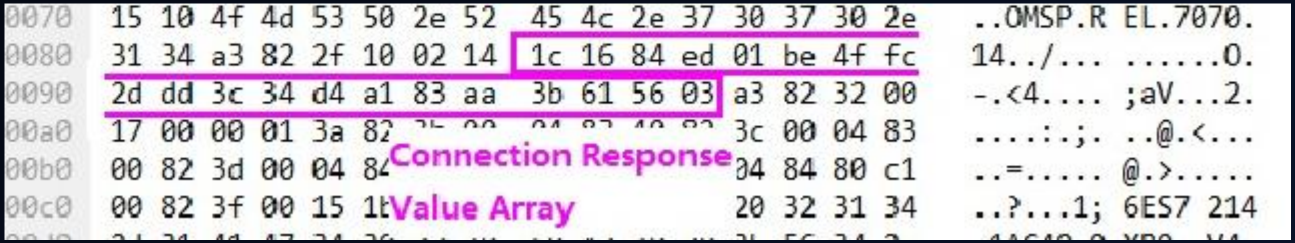
```
030  f6 6c b1 a3 00 00 03 00  00 65 02 f0 80 72 03 00   .l...... .e...r..
040  56 20 68 ad 71 74 34 cb  34 89 19 4d ae 03 0a d2   V h.qt4. 4..M....
050  e6 f5 7c 5e c3 07 a9 89  a5 5d 31 b0 c2 23 42 80   ..|^.... .]1..#B.
060  b8 fc 31 00 00 04 f2 00  00 00 0c 00 00 03 8f 34   ........ .......4
070  00 00 00 34 01 90 77 00  08 01 00 00 04 e8 89 69
```

Fun with the Encryption

- Using reverse debugging techniques, we found these encryption is calculated by TIA Portal through a file named OMSp_core_managed.dll

1. **Connection packet encryption**

Input parameter for this encryption is a random value array generated by the PLC in the first connection response packet.



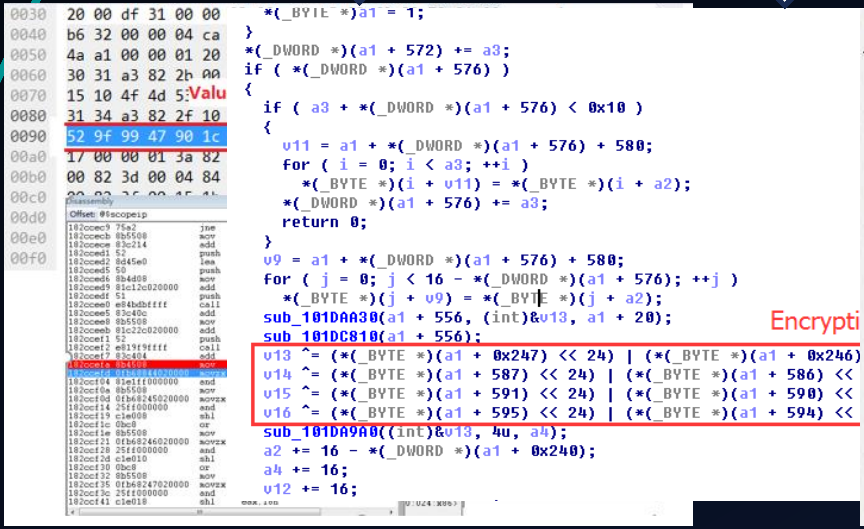
```
0070  15 10 4f 4d 53 50 2e 52  45 4c 2e 37 30 37 30 2e   ..OMSP.R EL.7070.
0080  31 34 a3 82 2f 10 02 14  1c 16 84 ed 01 be 4f fc   14../... ......O.
0090  2d dd 3c 34 d4 a1 83 aa  3b 61 56 03 a3 82 32 00   -.<4.... ;aV...2.
00a0  17 00 00 01 3a 82                    3c 00 04 83   ....:.;. ..@.<...
00b0  00 82 3d 00 04 84                    04 84 80 c1   ..=..... @.>.....
00c0  00 82 3f 00 15 1b                    20 32 31 34   ..?...1; 6ES7 214
```

Fun with the Encryption

（1） First encryption in the connection packet

Using XOR (we call this Encryption1), the first encryption can be calculated with the input parameter Value Array.

Value Array +Encryption1 = First Encryption



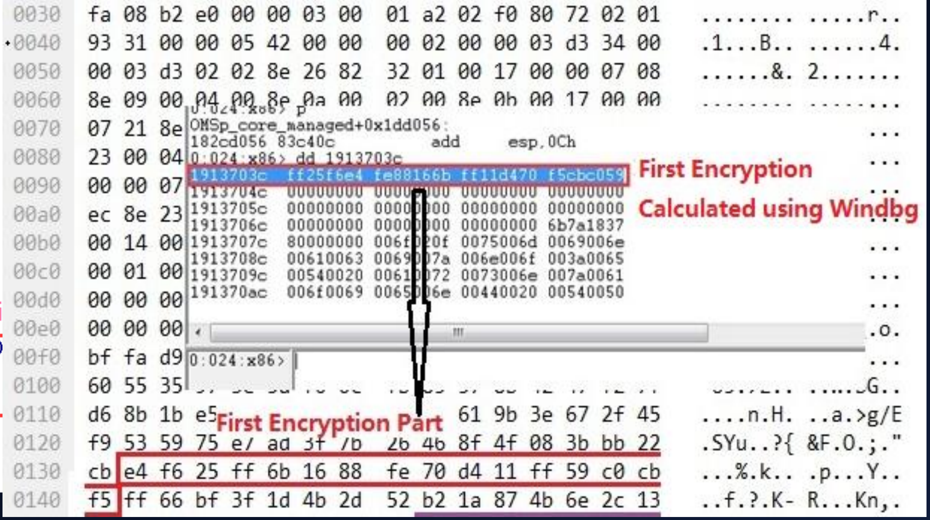
```
0030  20 00 df 31 00 00
0040  b6 32 00 00 04 ca
0050  4a a1 00 00 01 20
0060  30 31 a3 82 2b 00
0070  15 10 4f 4d 5
0080  31 34 a3 82 2f 10
0090  52 9f 99 47 90 1c
00a0  17 00 00 01 3a 82
00b0  00 82 3d 00 04 84
00c0
00d0
00e0
00f0
```

Valu

```
Disassembly
Offset: @$scopeip
182ccec9 75a2            jne
182ccecb 8b5508          mov
182ccece 83c214          add
182cced1 52              push
182cced2 8d45e0          lea
182cced5 50              push
182cced6 8b4d08          mov
182cced9 81c12c020000    add
182ccedf 51              push
182ccee0 e84bdbffff      call
182ccee5 83c40c          add
182ccee8 8b5508          mov
182cceeb 81c22c020000    add
182ccef1 52              push
182ccef2 e819f9ffff      call
182ccef7 83c404          add
182ccefa 8b4508          mov
182ccefd 0fb6884402000   movzx
182ccf04 81e1ff000000    and
182ccf0a 8b5508          mov
182ccf0d 0fb68245020000  movzx
182ccf14 25ff000000      and
182ccf19 c1e008          shl
182ccf1c 0bc8            or
182ccf1e 8b5508          mov
182ccf21 0fb68246020000  movzx
182ccf28 25ff000000      and
182ccf2d c1e010          shl
182ccf30 0bc8            or
182ccf32 8b5508          mov
182ccf35 0fb68247020000  movzx
182ccf3c 25ff000000      and
182ccf41 c1e018          shl
```

```
    *(_BYTE *)a1 = 1;
  }
  *(_DWORD *)(a1 + 572) += a3;
  if ( *(_DWORD *)(a1 + 576) )
  {
    if ( a3 + *(_DWORD *)(a1 + 576) < 0x10 )
    {
      v11 = a1 + *(_DWORD *)(a1 + 576) + 580;
      for ( i = 0; i < a3; ++i )
        *(_BYTE *)(i + v11) = *(_BYTE *)(i + a2);
      *(_DWORD *)(a1 + 576) += a3;
      return 0;
    }
    v9 = a1 + *(_DWORD *)(a1 + 576) + 580;
    for ( j = 0; j < 16 - *(_DWORD *)(a1 + 576); ++j )
      *(_BYTE *)(j + v9) = *(_BYTE *)(j + a2);
    sub_101DAA30(a1 + 556, (int)&v13, a1 + 20);
    sub_101DC810(a1 + 556);
    v13 ^= (*(_BYTE *)(a1 + 0x247) << 24) | (*(_BYTE *)(a1 + 0x246)
    v14 ^= (*(_BYTE *)(a1 + 587) << 24) | (*(_BYTE *)(a1 + 586) <<
    v15 ^= (*(_BYTE *)(a1 + 591) << 24) | (*(_BYTE *)(a1 + 590) <<
    v16 ^= (*(_BYTE *)(a1 + 595) << 24) | (*(_BYTE *)(a1 + 594) <<
    sub_101DA9A0((int)&v13, 4u, a4);
    a2 += 16 - *(_DWORD *)(a1 + 0x240);
    a4 += 16;
    v12 += 16;
```

Encrypti



```
0030  fa 08 b2 e0 00 00 03 00  01 a2 02 f0 80 72 02 01   ........ .....r..
0040  93 31 00 00 05 42 00 00  00 02 00 00 03 d3 34 00   .1...B.. ......4.
0050  00 03 d3 02 02 8e 26 82  32 01 00 17 00 00 07 08   ......&. 2.......
0060  8e 09 00 04 00 8e 0a 00  02 00 8e 0b 00 17 00 00   ........ ........
0070  07 21 8e                                                          ...
0080  23 00 04                                                          ...
0090  00 00 07                                                          ...
00a0  ec 8e 23                                                          ...
00b0  00 14 00                                                          ...
00c0  00 01 00                                                          ...
00d0  00 00 00                                                          ...
00e0  00 00 00                                                          .o.
00f0  bf fa d9                                                          ...
0100  60 55 35                                                    .....G..
0110  d6 8b 1b e5              61 9b 3e 67 2f 45   ....n.H. ..a.>g/E
0120  f9 53 59 75 e7 ad 3f 7b  26 46 8f 4f 08 3b bb 22   .SYu..?{ &F.O.;."
0130  cb e4 f6 25 ff 6b 16 88  fe 70 d4 11 ff 59 c0 cb   ...%.k.. .p...Y..
0140  f5 ff 66 bf 3f 1d 4b 2d  52 b2 1a 87 4b 6e 2c 13   ..f.?.K- R...Kn,.
```

```
0:024:x86> p
OMSp_core_managed+0x1dd056:
182cd056 83c40c          add     esp,0Ch
0:024:x86> dd 1913703c
1913703c  ff25f6e4 fe88166b ff11d470 f5cbc059
1913704c  00000000 00000000 00000000 00000000
1913705c  00000000 00000000 00000000 00000000
1913706c  00000000 00000000 00000000 6b7a1837
1913707c  80000000 006f020f 0075006d 0069006e
1913708c  00610063 0069007a 006e006f 003a0065
1913709c  00540020 00610072 0073006e 007a0061
191370ac  006f0069 0065006e 00440020 00540050
0:024:x86>
```

First Encryption

Calculated using Windbg

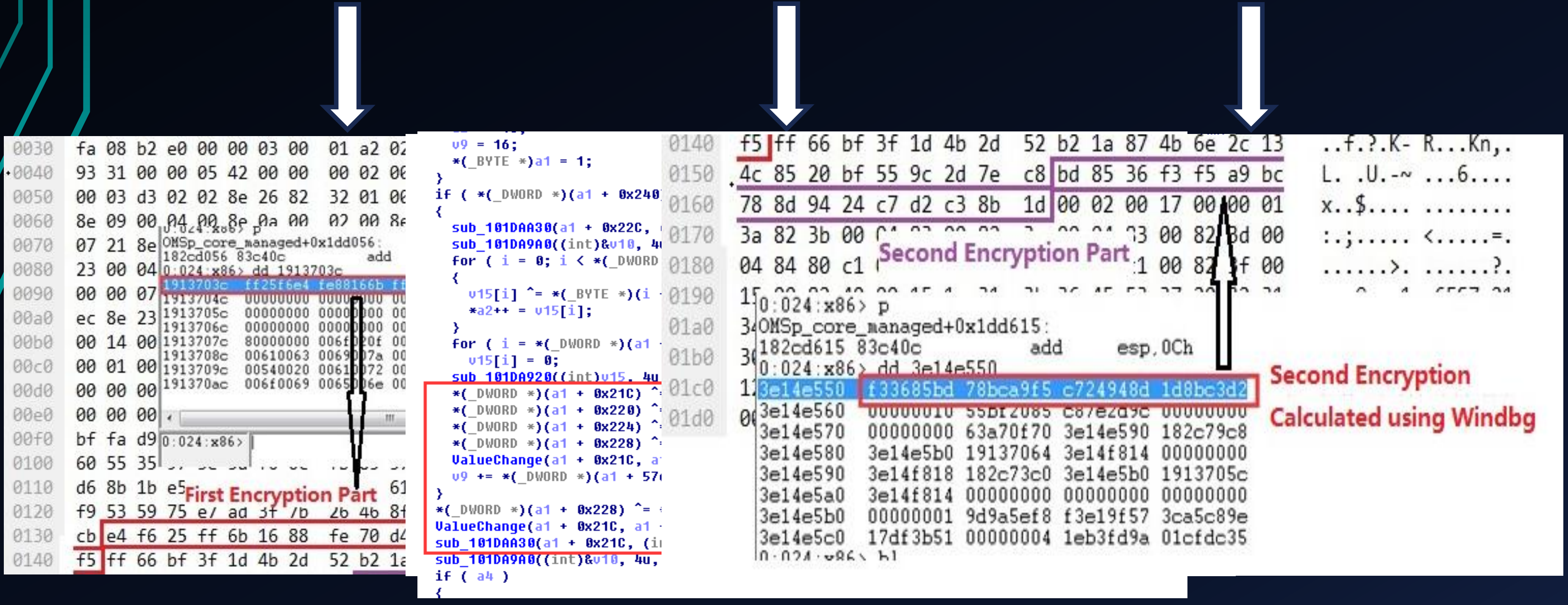
First Encryption Part

Fun with the Encryption

（2）Second encryption in the connection packet

Using the result of the first encryption as input parameter, the second encryption is calculated through a more complex Siemens-private algorithm.

First Encryption +Encryption2 = Second Encryption



```
0030  fa 08 b2 e0 00 00 03 00  01 a2 02
0040  93 31 00 00 05 42 00 00  00 02 00
0050  00 03 d3 02 02 8e 26 82  32 01 00
0060  8e 09 00 04 00 8e 0a 00  02 00 8e
0070  07 21 8e
0080  23 00 04
0090  00 00 07
00a0  ec 8e 23
00b0  00 14 00
00c0  00 01 00
00d0  00 00 00
00e0  00 00 00
00f0  bf fa d9
0100  60 55 35
0110  d6 8b 1b e5                      61
0120  f9 53 59 75 e7 ad 3f 7b  26 4b 8f
0130  cb e4 f6 25 ff 6b 16 88  fe 70 d4
0140  f5 ff 66 bf 3f 1d 4b 2d  52 b2 1a
```

```
0:024:x86> p
OMSp_core_managed+0x1dd056:
182cd056 83c40c          add
0:024:x86> dd 1913703c
1913703c  ff25f6e4 fe88166b ff
1913704c  00000000 00000000 00
1913705c  00000000 00000000 00
1913706c  00000000 00000000 00
1913707c  80000000 006f020f 00
1913708c  00610063 0069007a 00
1913709c  00540020 00610072 00
191370ac  006f0069 0065006e 00
0:024:x86>
```

First Encryption Part

```
    v9 = 16;
    *(_BYTE *)a1 = 1;
  }
  if ( *(_DWORD *)(a1 + 0x240)
  {
    sub_101DAA30(a1 + 0x22C,
    sub_101DA9A0((int)&v10, 4u
    for ( i = 0; i < *(_DWORD
    {
      v15[i] ^= *(_BYTE *)(i
      *a2++ = v15[i];
    }
    for ( i = *(_DWORD *)(a1
      v15[i] = 0;
    sub_101DA920((int)v15, 4u
    *(_DWORD *)(a1 + 0x21C) ^=
    *(_DWORD *)(a1 + 0x220) ^=
    *(_DWORD *)(a1 + 0x224) ^=
    *(_DWORD *)(a1 + 0x228) ^=
    ValueChange(a1 + 0x21C, a
    v9 += *(_DWORD *)(a1 + 57
  }
  *(_DWORD *)(a1 + 0x228) ^=
  ValueChange(a1 + 0x21C, a1
  sub_101DAA30(a1 + 0x21C, (i
  sub_101DA9A0((int)&v10, 4u,
  if ( a4 )
  {
```

```
0140  f5 ff 66 bf 3f 1d 4b 2d  52 b2 1a 87 4b 6e 2c 13   ..f.?.K- R...Kn,.
0150  4c 85 20 bf 55 9c 2d 7e  c8 bd 85 36 f3 f5 a9 bc   L. .U.-~ ...6....
0160  78 8d 94 24 c7 d2 c3 8b  1d 00 02 00 17 00 00 01   x..$.... ........
0170  3a 82 3b 00                    33 00 82 3d 00      :.;..... <.....=.
0180  04 84 80 c1                    1 00 82 3f 00       ......>. ......?.
0190  15
01a0  34
01b0  30
01c0  12
01d0  00
```

Second Encryption Part

```
0:024:x86> p
OMSp_core_managed+0x1dd615:
182cd615 83c40c          add     esp,0Ch
0:024:x86> dd 3e14e550
3e14e550  f33685bd 78bca9f5 c724948d 1d8bc3d2
3e14e560  00000010 55bf2085 c87e2d9c 00000000
3e14e570  00000000 63a70f70 3e14e590 182c79c8
3e14e580  3e14e5b0 19137064 3e14f814 00000000
3e14e590  3e14f818 182c73c0 3e14e5b0 1913705c
3e14e5a0  3e14f814 00000000 00000000 00000000
3e14e5b0  00000001 9d9a5ef8 f3e19f57 3ca5c89e
3e14e5c0  17df3b51 00000004 1eb3fd9a 01cfdc35
0:024:x86> bl
```

Second Encryption

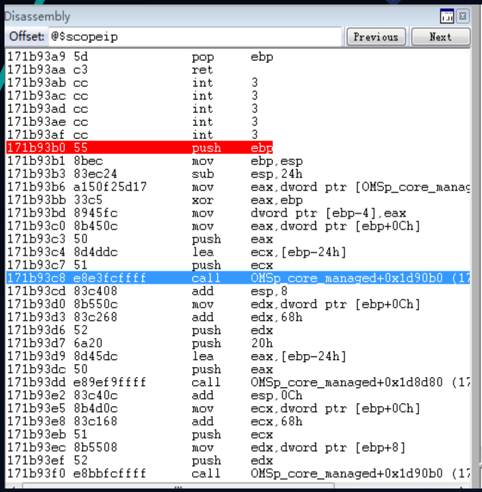
Calculated using Windbg

Fun with the Encryption

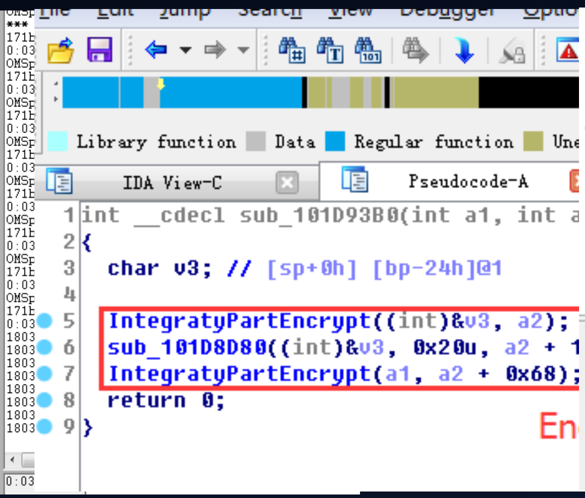
2. **Function packet encryption**

A fixed field array with Session ID is the input parameter. A complex algorithm (we call this Encryption3) is used to calculated the encryption result as follow:

ConstanArray (with Session ID) +Encryption3 = Function Encryption



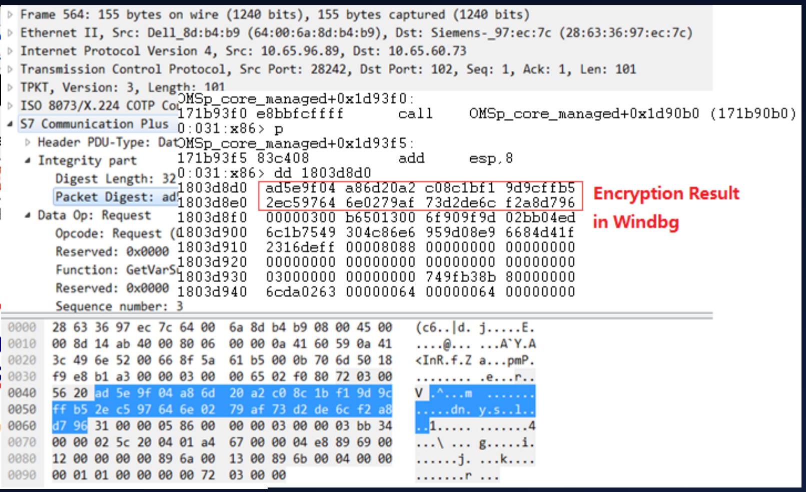
```
Offset: @$scopeip
171b93a9 5d            pop     ebp
171b93aa c3            ret
171b93ab cc            int     3
171b93ac cc            int     3
171b93ad cc            int     3
171b93ae cc            int     3
171b93af cc            int     3
171b93b0 55            push    ebp
171b93b1 8bec          mov     ebp,esp
171b93b3 83ec24        sub     esp,24h
171b93b6 a150f25d17    mov     eax,dword ptr [OMSp_core_manag
171b93bb 33c5          xor     eax,ebp
171b93bd 8945fc        mov     dword ptr [ebp-4],eax
171b93c0 8b450c        mov     eax,dword ptr [ebp+0Ch]
171b93c3 50            push    eax
171b93c4 8d4ddc        lea     ecx,[ebp-24h]
171b93c7 51            push    ecx
171b93c8 e8e3fcffff    call    OMSp_core_managed+0x1d90b0 (17
171b93cd 83c408        add     esp,8
171b93d0 8b550c        mov     edx,dword ptr [ebp+0Ch]
171b93d3 83c268        add     edx,68h
171b93d6 52            push    edx
171b93d7 6a20          push    20h
171b93d9 8d45dc        lea     eax,[ebp-24h]
171b93dc 50            push    eax
171b93dd e89ef9ffff    call    OMSp_core_managed+0x1d8d80 (17
171b93e2 83c40c        add     esp,0Ch
171b93e5 8b4d0c        mov     ecx,dword ptr [ebp+0Ch]
171b93e8 83c168        add     ecx,68h
171b93eb 51            push    ecx
171b93ec 8b5508        mov     edx,dword ptr [ebp+8]
171b93ef 52            push    edx
171b93f0 e8bbfcffff    call    OMSp_core_managed+0x1d90b0 (17
```



Library function ▨ Data 🟦 Regular function ▨ Une

```
int __cdecl sub_101D93B0(int a1, int a
{
  char v3; // [sp+0h] [bp-24h]@1

  IntegratyPartEncrypt((int)&v3, a2);
  sub_101D8D80((int)&v3, 0x20u, a2 + 1
  IntegratyPartEncrypt(a1, a2 + 0x68);
  return 0;
}
```



```
Frame 564: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits)
Ethernet II, Src: Dell_8d:b4:b9 (64:00:6a:8d:b4:b9), Dst: Siemens-_97:ec:7c (28:63:36:97:ec:7c)
Internet Protocol Version 4, Src: 10.65.96.89, Dst: 10.65.60.73
Transmission Control Protocol, Src Port: 28242, Dst Port: 102, Seq: 1, Ack: 1, Len: 101
TPKT, Version: 3, Length: 101
ISO 8073/X.224 COTP Co
S7 Communication Plus
  Header PDU-Type: Dat
  Integrity part
    Digest Length: 32
    Packet Digest: ad
  Data Op: Request
    Opcode: Request (
    Reserved: 0x0000
    Function: GetVarS
    Reserved: 0x0000
    Sequence number: 3
```

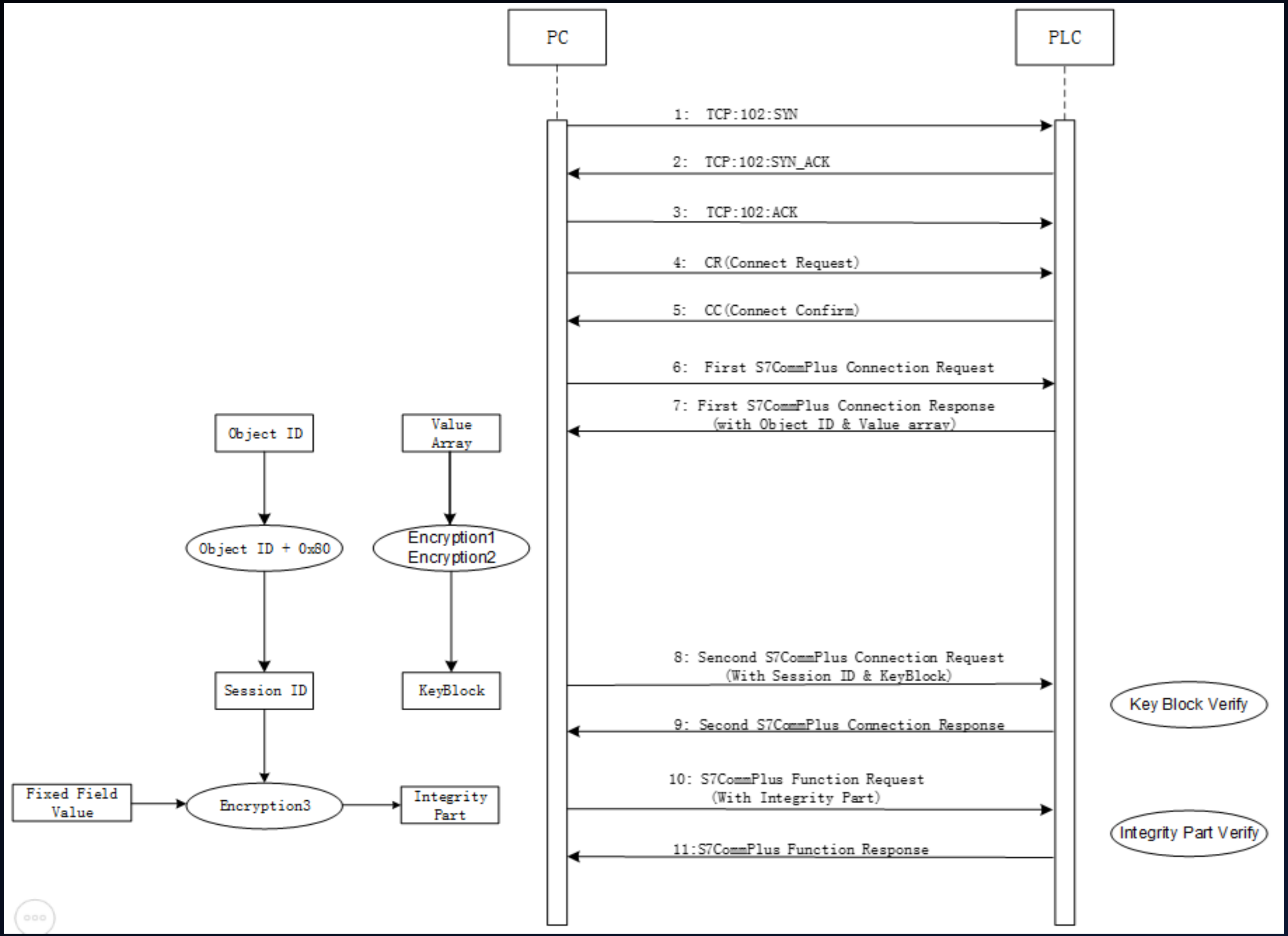
```
OMSp_core_managed+0x1d93f0:
171b93f0 e8bbfcffff      call    OMSp_core_managed+0x1d90b0 (171b90b0)
0:031:x86> p
OMSp_core_managed+0x1d93f5:
171b93f5 83c408          add     esp,8
0:031:x86> dd 1803d8d0
1803d8d0  ad5e9f04 a86d20a2 c08c1bf1 9d9cffb5
1803d8e0  2ec59764 6e0279af 73d2de6c f2a8d796
1803d8f0  00000300 b6501300 6f909f9d 02bb04ed
1803d900  6c1b7549 304c86e6 959d08e9 6684d41f
1803d910  2316deff 00008088 00000000 00000000
1803d920  00000000 00000000 00000000 00000000
1803d930  03000000 00000000 749fb38b 80000000
1803d940  6cda0263 00000064 00000064 00000000
```

Encryption Result in Windbg

```
0000  28 63 36 97 ec 7c 64 00  6a 8d b4 b9 08 00 45 00   (c6..|d. j.....E.
0010  00 8d 14 ab 40 00 80 06  00 00 0a 41 60 59 0a 41   ....@... ...A`Y.A
0020  3c 49 6e 52 00 66 8f 5a  61 b5 00 0b 70 6d 50 18   <InR.f.Z a...pmP.
0030  f9 e8 b1 a3 00 00 03 00  00 65 02 f0 80 72 03 00   ........ .e...r..
0040  56 20 ad 5e 9f 04 a8 6d  20 a2 c0 8c 1b f1 9d 9c   V .^...m  .......
0050  ff b5 2e c5 97 64 6e 02  79 af 73 d2 de 6c f2 a8   .....dn. y.s..l..
0060  d7 96 31 00 00 05 86 00  00 00 03 00 00 03 bb 34   ..1..... .......4
0070  00 00 02 5c 20 04 01 a4  67 00 00 04 e8 89 69 00   ...\ ... g.....i.
0080  12 00 00 00 00 89 6a 00  13 00 89 6b 00 04 00 00   ......j. ...k....
0090  00 01 01 00 00 00 00 72  03 00 00                  .......r ...
```

Fun with the Encryption

3. **S7CommPlus Communication with Encryption**


PC
PLC
1: TCP:102:SYN
2: TCP:102:SYN_ACK
3: TCP:102:ACK
4: CR(Connect Request)
5: CC(Connect Confirm)
6: First S7CommPlus Connection Request
7: First S7CommPlus Connection Response (with Object ID & Value array)
Object ID
Value Array
Object ID + 0x80
Encryption1 Encryption2
Session ID
KeyBlock
Fixed Field Value
Encryption3
Integrity Part
8: Sencond S7CommPlus Connection Request (With Session ID & KeyBlock)
Key Block Verify
9: Second S7CommPlus Connection Response
10: S7CommPlus Function Request (With Integrity Part)
Integrity Part Verify
11:S7CommPlus Function Response

Protections

- **Code level**:

  -- Use code confusion techniques and anti-Debug techniques for the key DLL files

- **Design level**

  -- use a private key as an input parameter for encryption algorithm in the communication between Siemens software and PLCs.

- **Protocol level**

  -- Encrypt the whole packets instead of the key byte encryption

Thank You!

chengleim19@gmail.com