PacSec 2008 Conference

# Inside "Winnyp" - Winnyp Internals and Concepts of Network Crawling

Fourteenforty Research Institute, Inc.
http://www.fourteenforty.jp

Senior Software Engineer
Toshiaki Ishiyama

## Summary

- Winnyp is an anonymous P2P filesharing software based on Winny.
- Winnyp is compatible with Winny, it can communicate with Winny.
- Encryption key generation algorithm of Winnyp is more difficult than Winny .
- The report of having analyzed Winnyp has not gone up up to now.
- I report on the analytical result of the cryptographic algorithm of Winnyp and the outline of crawling system (WinnypRadar).

## Agenda

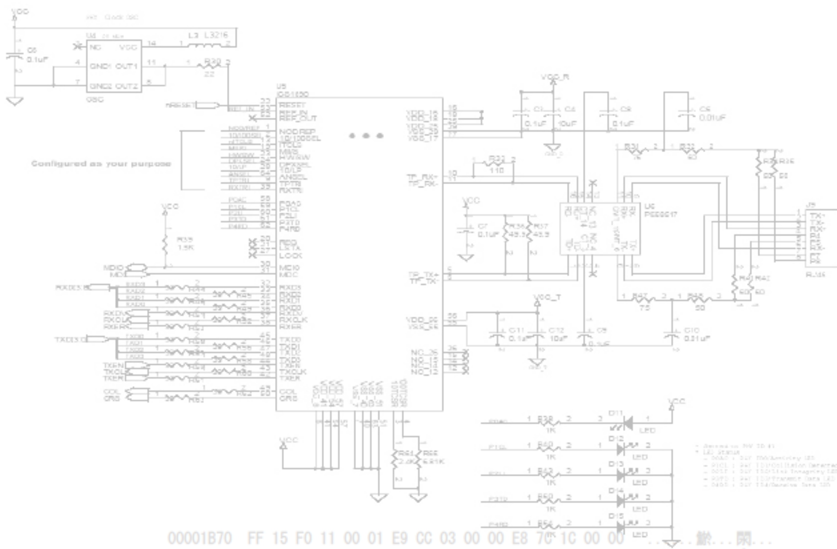
1. Internal Winnyp.

2. The approaches to the statical analysis for anonymous P2P filesharing systems.

3. WinnypRadar – Winnyp network crawler-

# 1. Internal Winnyp

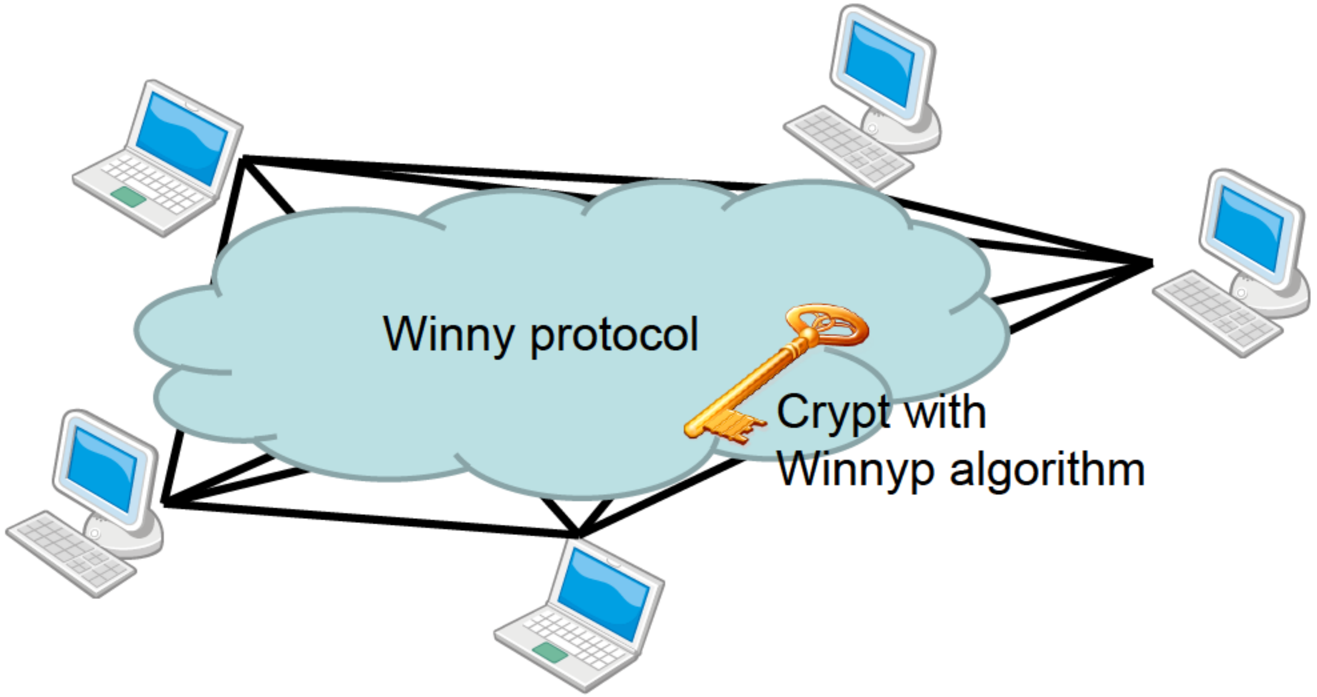
```
int packet_analysis(GDDCONFIG *gddc,unsigned char *packet,unsigned long length)
{
    struct ip        *ip_header;     /* IP header  */
    struct tcphdr    *tcp_header;    /* TCP header */
    char             *tcp_data;      /* TCP data */
    struct in_addr   addr;           /* IP address */
    char             sourceIP[16];   /* Source IP address */
    char             destIP[16];     /* Destination IP address  */
    unsigned short   sourcePort;     /* Source Port */
    unsigned short   destPort;       /* Destination Port */
    unsigned long    len_data;       /* Length of data part */
    unsigned long    iph_len;        /* Length of IP header */
    unsigned long    tcph_len;       /* Length of TCP header */
    unsigned long    sequence;       /* Expected sequence */
    [illegible]      por[illegible]ndex;     /* Indexnumber of port lis[illegible]*/
    in[illegible]    dir[illegible]
    un[illegible]igned char    log[illegible]
    C[illegible]     *bo[illegible]
    CONN_LIST        *t;             /* Temporary connection list */
    static char      datestr[512];   /* Buffer to store datetime */
    time_t           timeval;
    struct tm        *timep=NULL;
    char             *timesp=NULL;
    char             *c;

    /* Get pointer of IP header and check length of IP */
    if (length-SIZE_OF_ETHHDR < MINSIZE_IP+MINSIZE_TCP) return(0);
    ip_header   = (struct ip *)(packet+SIZE_OF_ETHHDR);
    if (ip_header->ip_p!=IPPROTO_TCP
     || ip_header->ip_v!=4) return(0);
    iph_len     = ((unsigned long)(ip_header->ip_hl))*4;
    if (iph_len<MINSIZE_IP) return(0);
    if ((unsigned long)ntohs(ip_header->ip_len) < MINSIZE_IP+MINSIZE_TCP)
        return(0);
    if ((unsigned long)ntohs(ip_header->ip_len) > length-SIZE_OF_ETHHDR){
        return(0);
    }

    /* Get pointer of TCP header and check length of TCP */
    tcp_header  = (struct tcphdr *)((char *)ip_header+iph_len);
    tcph_len    = ((unsigned long)(tcp_header->th_off))*4;
    tcp_data    = (char *)tcp_header+tcph_len;
    if (tcph_len<MINSIZE_TCP) return(0);

    /* Get other parameter in TCP/IP header */
    if ((long)ntohs(ip_header->ip_len)-(long)iph_len-(long)tcph_len<0)
        return(0);
    len_data    = (unsigned long)ntohs(ip_header->ip_len)
                     -iph_len-tcph_len;
    sourcePort  = ntohs(tcp_header->th_sport);
    destPort    = ntohs(tcp_header->th_dport);
    memcpy(&addr,&(ip_header->ip_src),sizeof(struct in_addr));
    strcpy(sourceIP,(char *)inet_ntoa(addr));
    memcpy(&addr,&(ip_header->ip_dst),sizeof(struct in_addr));
    strcpy(destIP,(char *)inet_ntoa(addr));
    if (!strcmp(sourceIP,destIP)) return(0);
```



```
00001B70  FF 15 F0 11 00 01 E9 CC 03 00 00 E8 7C 1C 00 00   ......齡...閑...
00001B80  33 F6 56 E8 B1 FC FF FF 85 C0 0F 84 B7 03 00 00   3・臙...・.┤...
00001B90  56 6A 02 FF 35 6C 80 00 01 FF 35 D0 87 00 01 FF   Vj..5l....5ミ....
00001BA0  15 CC 11 00 01 85 C0 75 1D 68 10 10 00 00 FF 35   .ﾌ...・u.h.....5
00001BB0  50 80 00 01 FF 35 44 80 00 01 FF 35 D0 87 00 01   P....5D....5ミ...
00001BC0  FF 15 04 12 00 01 FF 35 D0 87 00 01 FF 15 2C 12   .......5ミ.....,.
00001BD0  00 01 FF 35 D4 8B 00 01 FF 15 58 10 00 01 E9 64   ...5ﾔ.....X...鍼
00001BE0  03 00 00 83 FE 1A 77 47 0F 84 59 03 00 00 83 FE   ......wG.ㇽ.....
00001BF0  11 0F 85 16 01 00 00 33 F6 39 35 E8 87 00 01 74   .......3.95闕..t
00001C00  22 8B 3D 28 12 00 01 56 FF D7 56 FF D7 68 00 10   ".=(...V.ﾗV.ﾗh..
00001C10  00 00 FF 35 50 80 00 01 FF 35 88 80 00 01 E9 7D   ...5P....5・..驀
00001C20  02 00 00 6A 01 E8 0F FC FF FF E9 1A 03 00 00 8B   ...j...........急
00001C30  7D 14 B8 11 01 00 00 3B F0 0F 87 8B 00 00 00 3B   .ｸ....;..㈲...;
00001C40  F0 0F 84 16 02 00 00 83 FE 1C 0F 85 BD 00 00 00   ............・...
00001C50  33 F6 39 75 10 74 2F A1 EC 87 00 01 8B 0D F0 87   3.9u.t/｡・.....・
00001C60  00 01 3B C6 75 08 3B CE 0F 84 D9 02 00 00 8B 3D   ..;ﾆu.;ﾎ.・....=
00001C70  14 12 00 01 51 50 68 B1 00 00 00 FF 35 D4 87 00   ....QPhｱ....5ﾔ..
00001C80  01 E9 56 01 00 00 8B 3D 14 12 00 01 68 F0 87 00   .鍋....=....h・.
00001C90  01 68 EC 87 00 01 68 B0 00 00 00 FF 35 D4 87 00   .h・..hｰ....5ﾔ..
00001CA0  01 FF D7 A1 EC 87 00 01 8B 0D F0 87 00 01 3B C1   ..ﾗ｡・.....・..;ﾁ
00001CB0  75 11 89 35 EC 87 00 01 89 35 F0 87 00 01 E9 84   u..5・....5・..驗
00001CC0  02 00 00 51 50 E9 07 01 00 00 8B CE B8 12 01 00   ...QP.....動ｸ...
00001CD0  00 2B C8 0F 84 3C 02 00 00 83 E9 04 0F 84 29 02   .+ﾈ..<...・...).
00001CE0  00 00 49 0F 84 F9 01 00 00 81 E9 1C 01 00 00 0F   ..I.・....・.....
00001CF0  84 E0 01 00 00 81 E9 E6 00 00 00 0F 84 3D 01 00   ・....・......=..
00001D00  00 81 E9 E8 7C 00 00 0F 84 02 01 00 00 3B 35 5C   .・閑........;5\
00001D10  88 00 01 0F 85 EE 00 00 00 8B 45 14 8B 48 0C 8B   ....・...畿.稀.驚
00001D20  C1 8B D1 F7 D0 C1 EA 02 83 E0 01 83 E2 01 F6 C1   錦・ﾁ..・..・..・
```

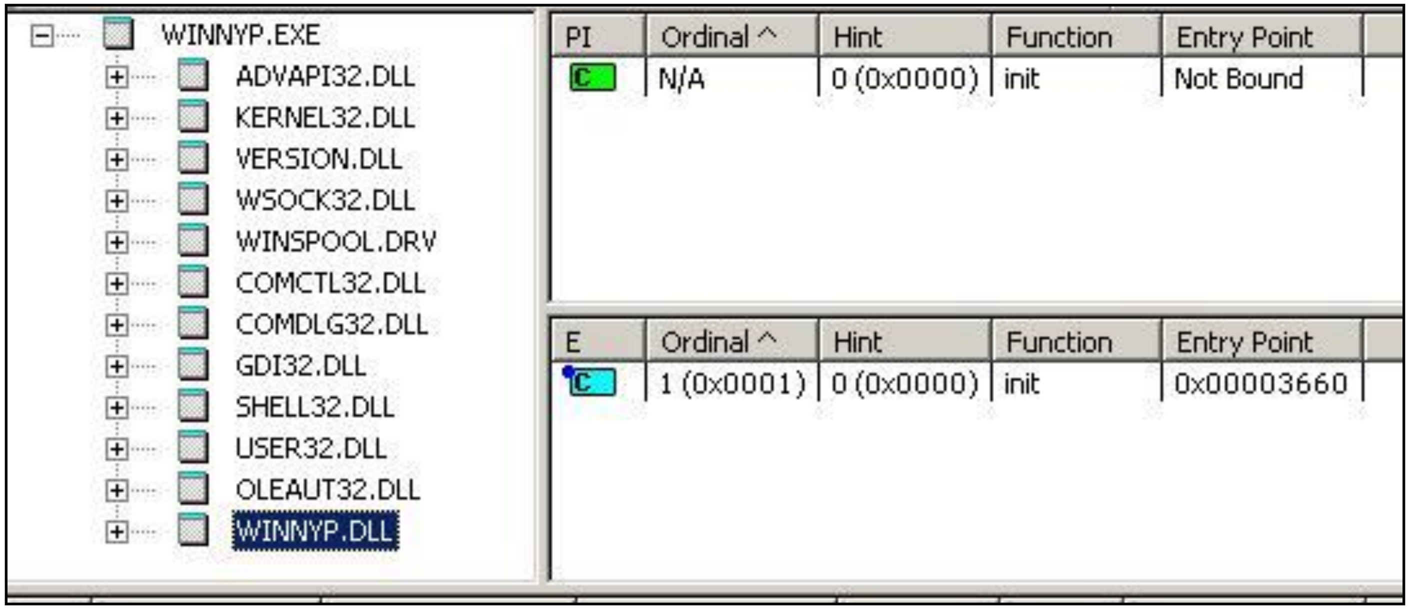
## Outline of Winnyp

- Winnyp is an anonymous P2P filesharing software based on Winny.
- Winnyp is compatible with Winny. It communicates with Winny protocol. But Winnyp uses original algorithm in encryption key generation process.



## Outline of Winnyp

- Winnyp lets the executable file of Winny read winnyp.dll.
- Modified file calls the init function of winnyp.dll.



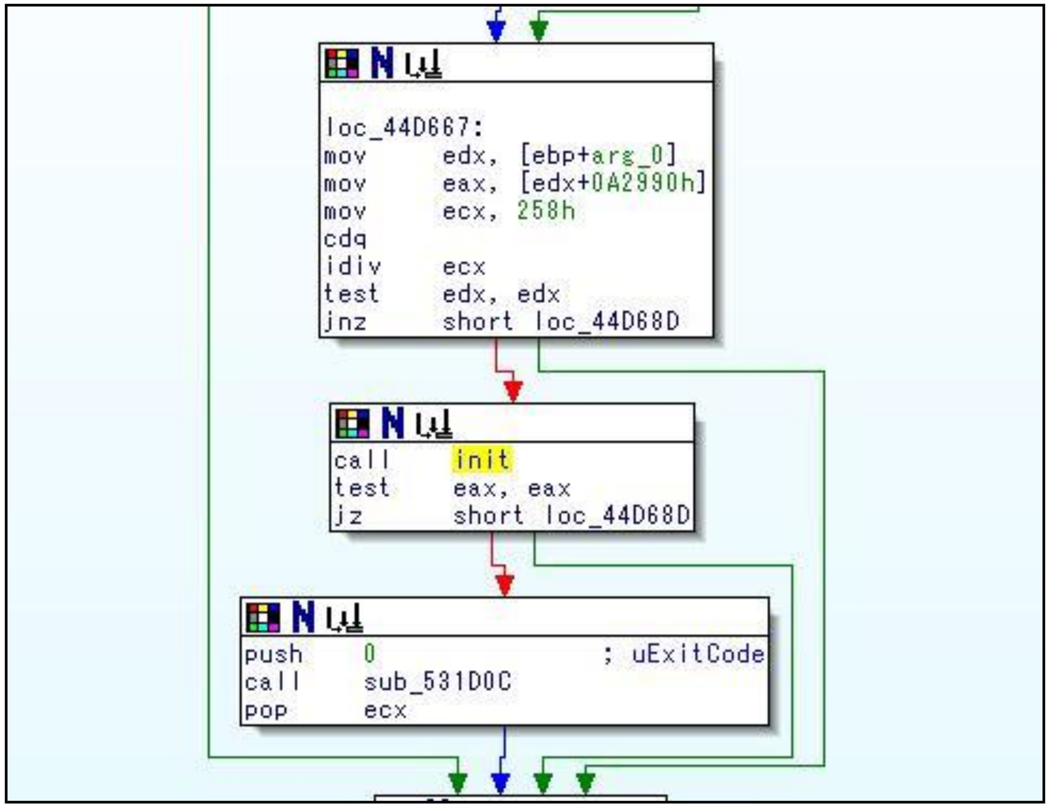
```
WINNYP.EXE
  ADVAPI32.DLL
  KERNEL32.DLL
  VERSION.DLL
  WSOCK32.DLL
  WINSPOOL.DRV
  COMCTL32.DLL
  COMDLG32.DLL
  GDI32.DLL
  SHELL32.DLL
  USER32.DLL
  OLEAUT32.DLL
  WINNYP.DLL
```

| PI | Ordinal ^ | Hint | Function | Entry Point |
|---|---|---|---|---|
| C | N/A | 0 (0x0000) | init | Not Bound |

| E | Ordinal ^ | Hint | Function | Entry Point |
|---|---|---|---|---|
| C | 1 (0x0001) | 0 (0x0000) | init | 0x00003660 |

## Outline of Winnyp

- Rewrite instruction for calling init function



```
loc_44D667:
mov     edx, [ebp+arg_0]
mov     eax, [edx+0A2990h]
mov     ecx, 258h
cdq
idiv    ecx
test    edx, edx
jnz     short loc_44D68D
```

```
call    init
test    eax, eax
jz      short loc_44D68D
```

```
push    0               ; uExitCode
call    sub_531D0C
pop     ecx
```

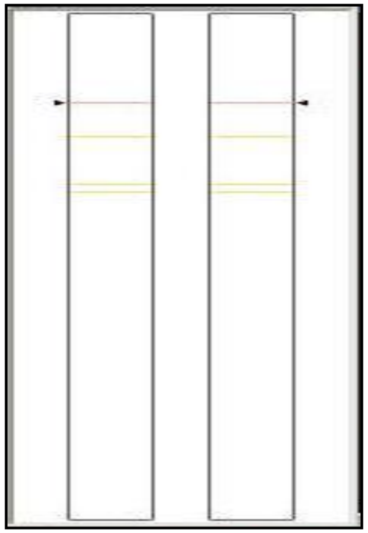
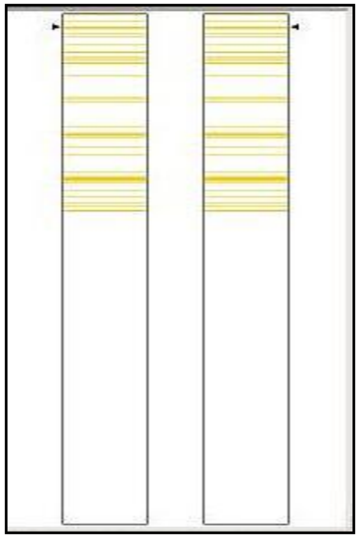
## Initialization of Winnyp

- Read configuraton file for Winnyp.（disper.ini）
- Create parameter for generating encryption key
- Create parameter for sending packet
- Patching Winnyp.exe

## Initialization of Winnyp

- In the patch processing to Winnyp.exe, the rewriting processing in about 200 places is executed.

- The majority of the rewriting processing are changes in the referred character string.   ex) Noderef.txt  > Noderefp.txt

## Encryption key generation algorithm

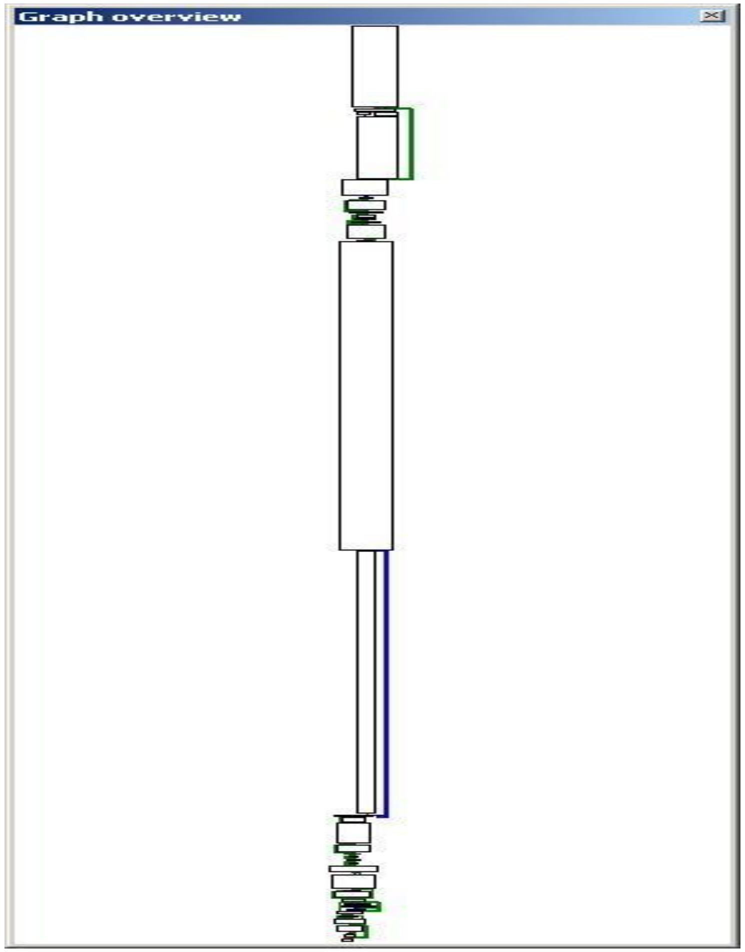
- In Winny, we could analyze the cryptographic algorithm easily. Because it's easy.

- In Winnyp, the cryptographic algorithm is RC4. However, the encryption key generation algorithm combines two or more algorithms, and the analysis is difficult.

In Winnyp
Encryption key
generation algorithm
?????

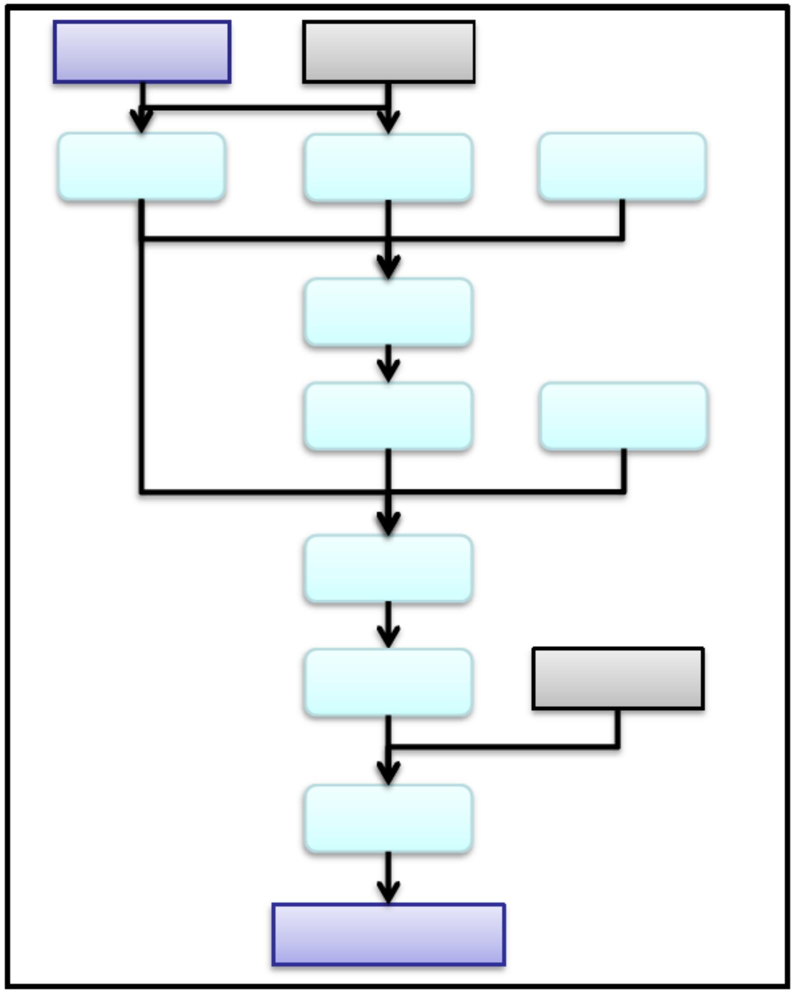
Winny
Encryption key
generation algorithm
RC4

## Encryption key generation algorithm – Outline –

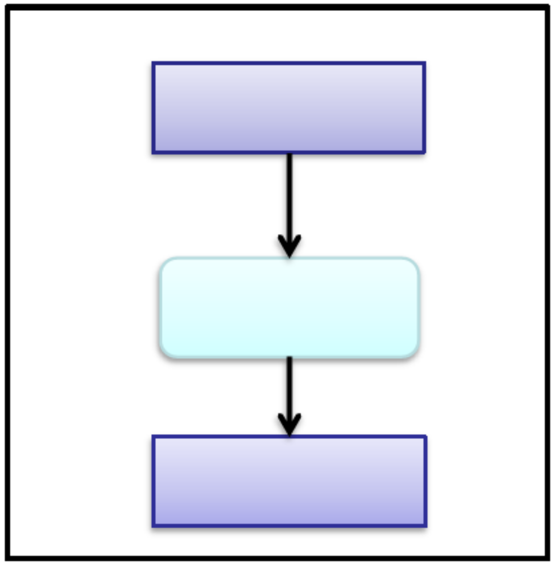


- Winnyp’s main routine for generating encryption key
- Very complex, very long

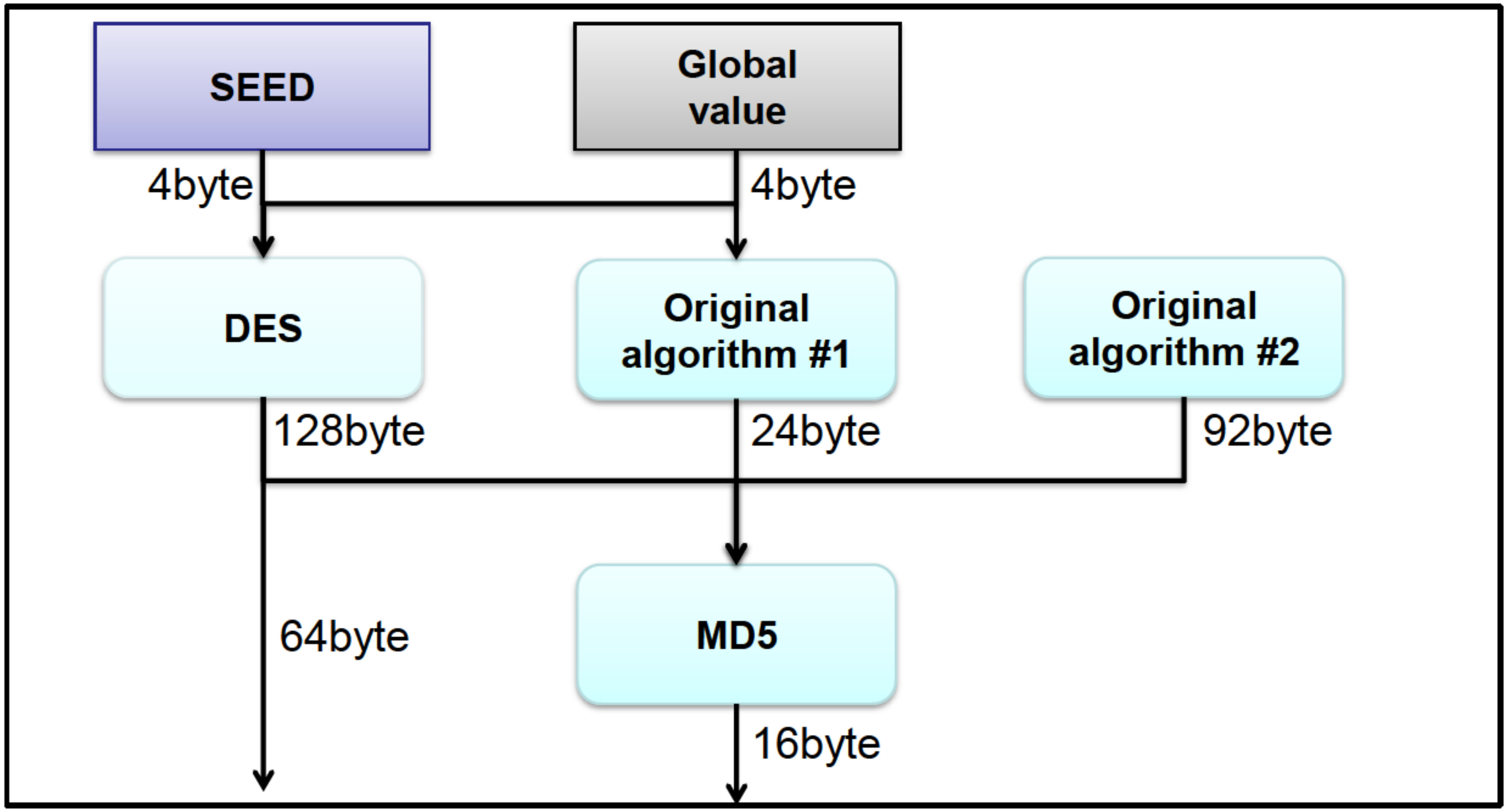
## Encryption key generation algorithm – Outline –



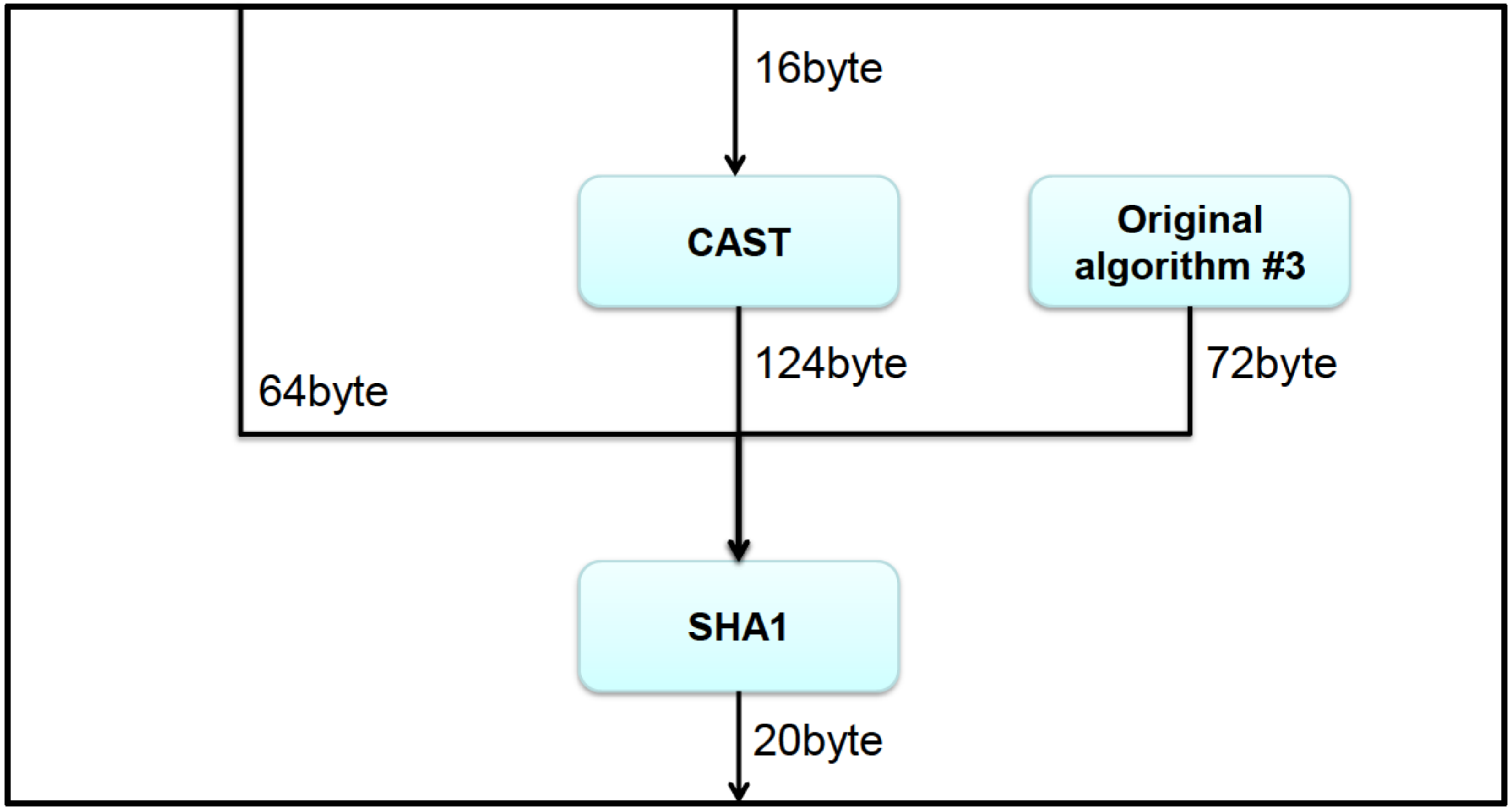
- The flow which divided by the processing block.
- More complex than Winny.



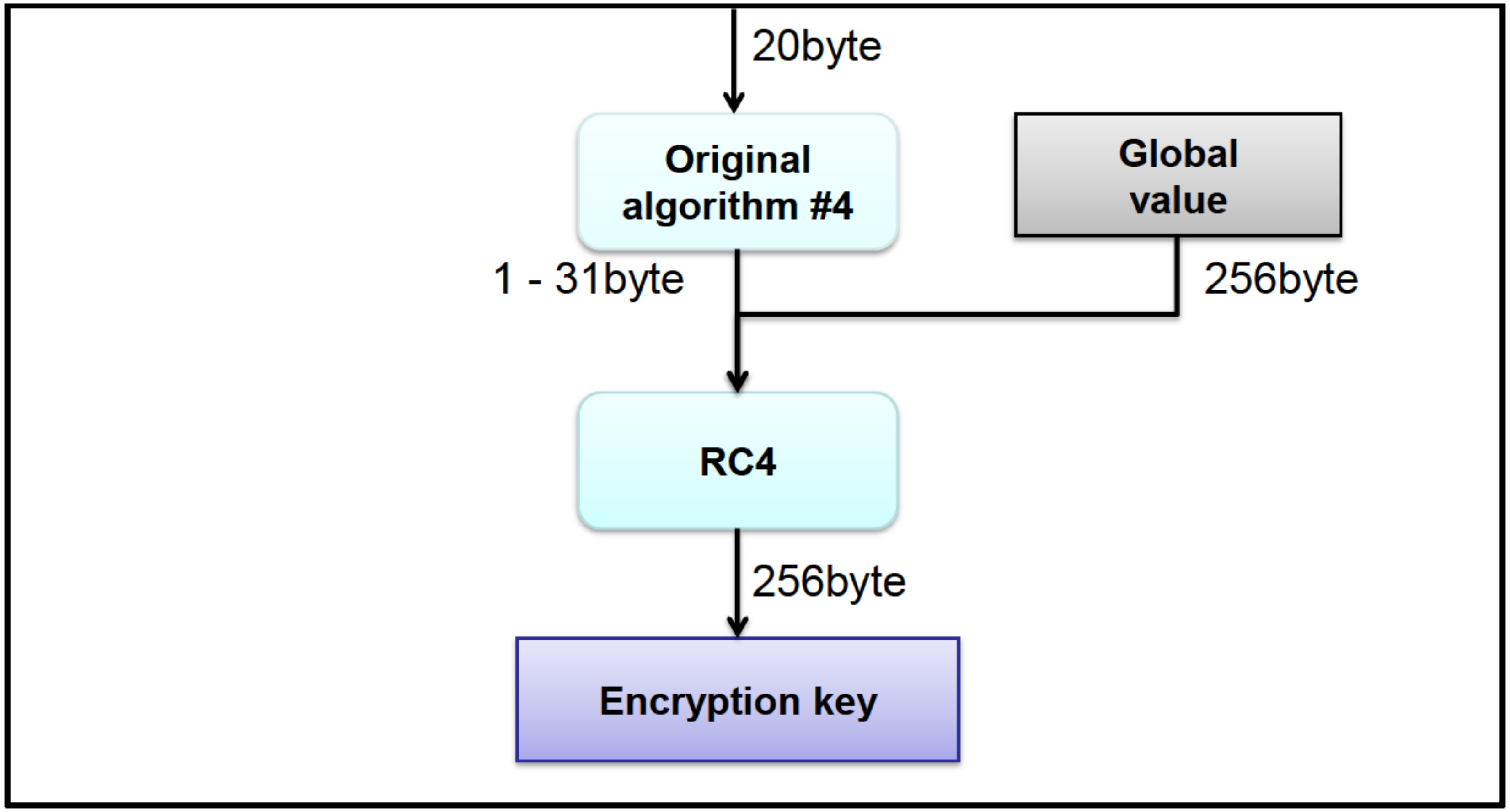
## Encryption key generation algorithm – Stage 1 –



## Encryption key generation algorithm – Stage 2 –



## Encryption key generation algorithm – Stage 3 –



## Sending packet

- In winnyp’s packet sending process, add dummy data after winny packet.
- Encryption key generation algorithm is chosen by Winnyp configuration file. (for Winny or Winnyp)
- Dummy data are added when kind of packet is connection establishing or connection closing.

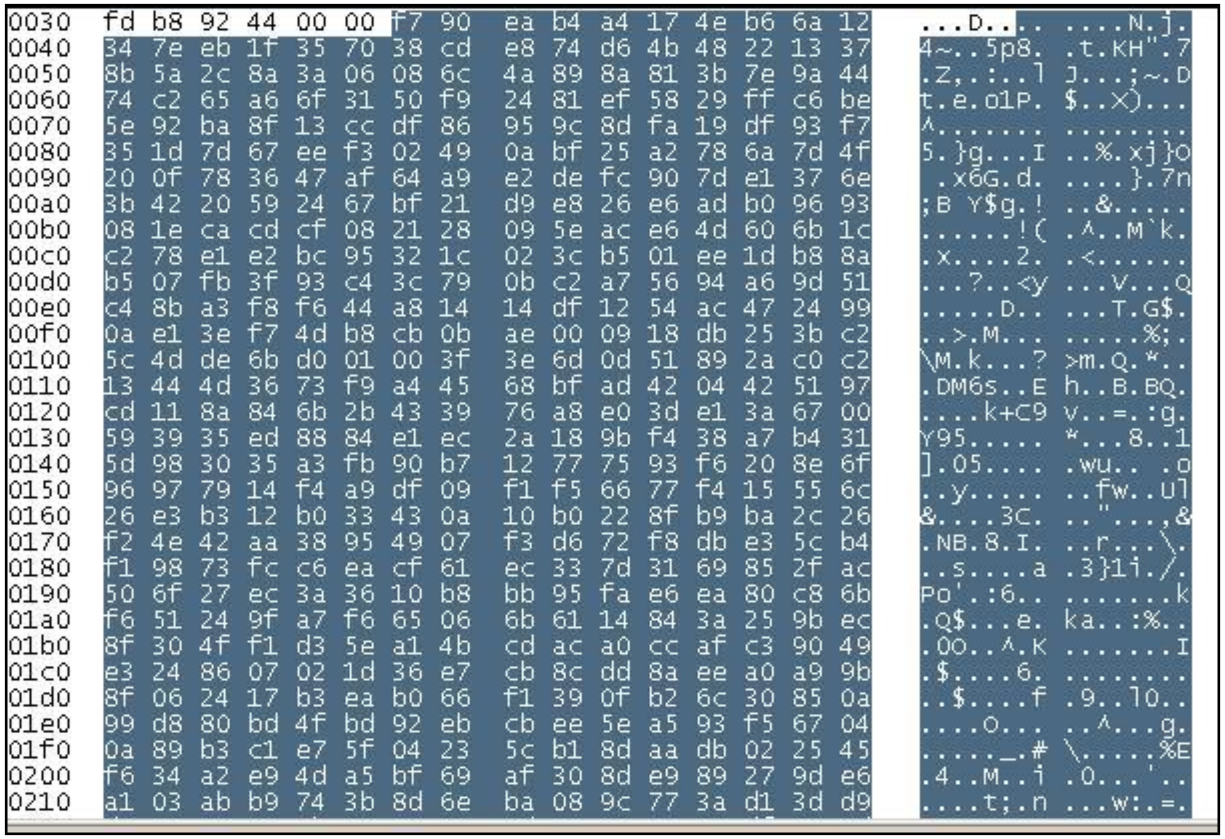
| | |
|---|---|
| **Length(4byte)** | **Data** |

| | | |
|---|---|---|
| **Length(4byte)** | **Dara** | **Dummy** |

FFR

## Sending packet

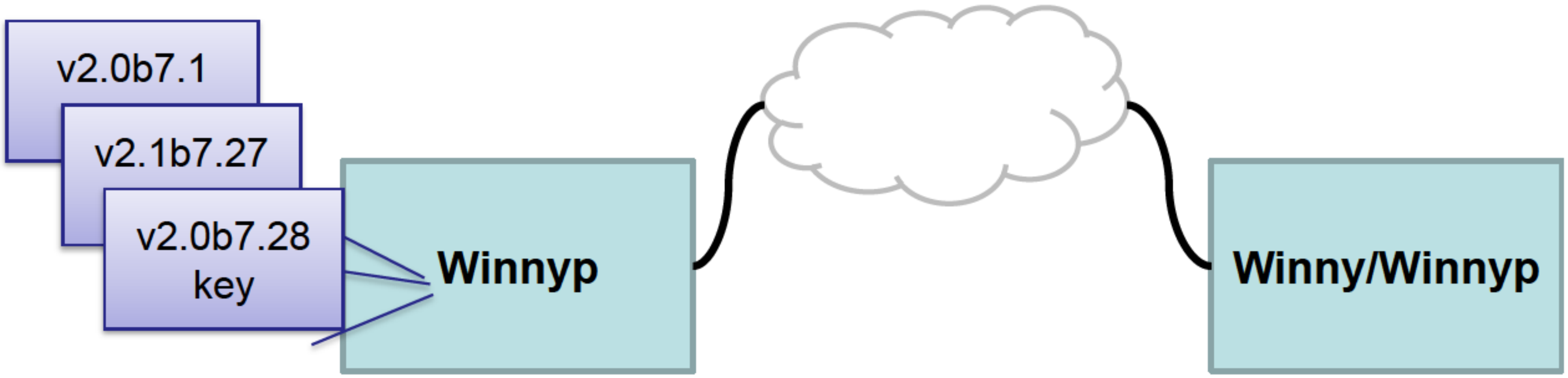
- Initial packet with dummy bytes



```
0030   fd b8 92 44 00 00 f7 90  ea b4 a4 17 4e b6 6a 12   ...D.... ....N.j.
0040   34 7e eb 1f 35 70 38 cd  e8 74 d6 4b 48 22 13 37   4~..5p8. .t.KH".7
0050   8b 5a 2c 8a 3a 06 08 6c  4a 89 8a 81 3b 7e 9a 44   .Z,.:..l J...;~.D
0060   74 c2 65 a6 6f 31 50 f9  24 81 ef 58 29 ff c6 be   t.e.o1P. $..X)...
0070   5e 92 ba 8f 13 cc df 86  95 9c 8d fa 19 df 93 f7   ^....... ........
0080   35 1d 7d 67 ee f3 02 49  0a bf 25 a2 78 6a 7d 4f   5.}g...I ..%.xj}O
0090   20 0f 78 36 47 af 64 a9  e2 de fc 90 7d e1 37 6e    .x6G.d. ....}.7n
00a0   3b 42 20 59 24 67 bf 21  d9 e8 26 e6 ad b0 96 93   ;B Y$g.! ..&.....
00b0   08 1e ca cd cf 08 21 28  09 5e ac e6 4d 60 6b 1c   ......!( .^..M`k.
00c0   c2 78 e1 e2 bc 95 32 1c  02 3c b5 01 ee 1d b8 8a   .x....2. .<......
00d0   b5 07 fb 3f 93 c4 3c 79  0b c2 a7 56 94 a6 9d 51   ...?..<y ...V...Q
00e0   c4 8b a3 f8 f6 44 a8 14  14 df 12 54 ac 47 24 99   .....D.. ...T.G$.
00f0   0a e1 3e f7 4d b8 cb 0b  ae 00 09 18 db 25 3b c2   ..>.M... .....%;.
0100   5c 4d de 6b d0 01 00 3f  3e 6d 0d 51 89 2a c0 c2   \M.k...? >m.Q.*..
0110   13 44 4d 36 73 f9 a4 45  68 bf ad 42 04 42 51 97   .DM6s..E h..B.BQ.
0120   cd 11 8a 84 6b 2b 43 39  76 a8 e0 3d e1 3a 67 00   ....k+C9 v..=.:g.
0130   59 39 35 ed 88 84 e1 ec  2a 18 9b f4 38 a7 b4 31   Y95..... *...8..1
0140   5d 98 30 35 a3 fb 90 b7  12 77 75 93 f6 20 8e 6f   ].05.... .wu.. .o
0150   96 97 79 14 f4 a9 df 09  f1 f5 66 77 f4 15 55 6c   ..y..... ..fw..Ul
0160   26 e3 b3 12 b0 33 43 0a  10 b0 22 8f b9 ba 2c 26   &....3C. .."...,&
0170   f2 4e 42 aa 38 95 49 07  f3 d6 72 f8 db e3 5c b4   .NB.8.I. ..r...\.
0180   f1 98 73 fc c6 ea cf 61  ec 33 7d 31 69 85 2f ac   ..s....a .3}1i./.
0190   50 6f 27 ec 3a 36 10 b8  bb 95 fa e6 ea 80 c8 6b   Po'.:6.. .......k
01a0   f6 51 24 9f a7 f6 65 06  6b 61 14 84 3a 25 9b ec   .Q$...e. ka..:%..
01b0   8f 30 4f f1 d3 5e a1 4b  cd ac a0 cc af c3 90 49   .0O..^.K .......I
01c0   e3 24 86 07 02 1d 36 e7  cb 8c dd 8a ee a0 a9 9b   .$....6. ........
01d0   8f 06 24 17 b3 ea b0 66  f1 39 0f b2 6c 30 85 0a   ..$....f .9..l0..
01e0   99 d8 80 bd 4f bd 92 eb  cb ee 5e a5 93 f5 67 04   ....O... ..^...g.
01f0   0a 89 b3 c1 e7 5f 04 23  5c b1 8d aa db 02 25 45   ....._.# \.....%E
0200   f6 34 a2 e9 4d a5 bf 69  af 30 8d e9 89 27 9d e6   .4..M..i .0...'..
0210   a1 03 ab b9 74 3b 8d 6e  ba 08 9c 77 3a d1 3d d9   ....t;.n ...w:.=.
```

## Receiving packet

- Winnyp creates three encryption keys when initialization packet received.（Winny v2.0b7.1, Winnyp v2.1b7.27, Winnyp v2.1b7.28）
- Specifies the version of connected node by using each keys.
- Winnyp can communicate with multiple version by these feature.



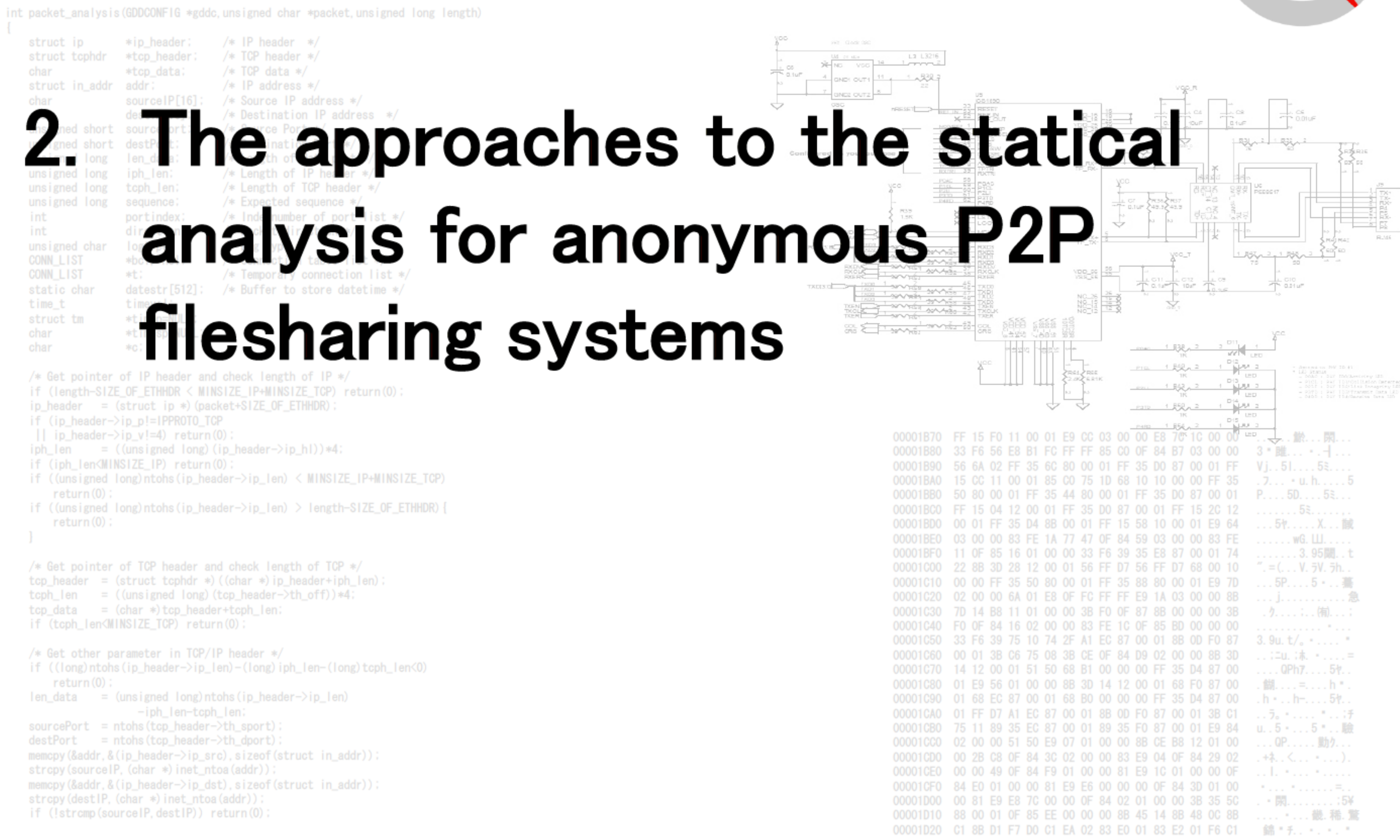
## Specifying target node

- Decrypt the first 5 bytes.
- Some checks are done to decrypted 5 bytes.

| | |
|---|---|
| Check #1 | Length is greater than 0 |
| Check #2 | Length is less than 131,072 |
| Check #3 | Length plus 4 is less than the receiving packet length |
| Check #4 | Command number is less than 100 |

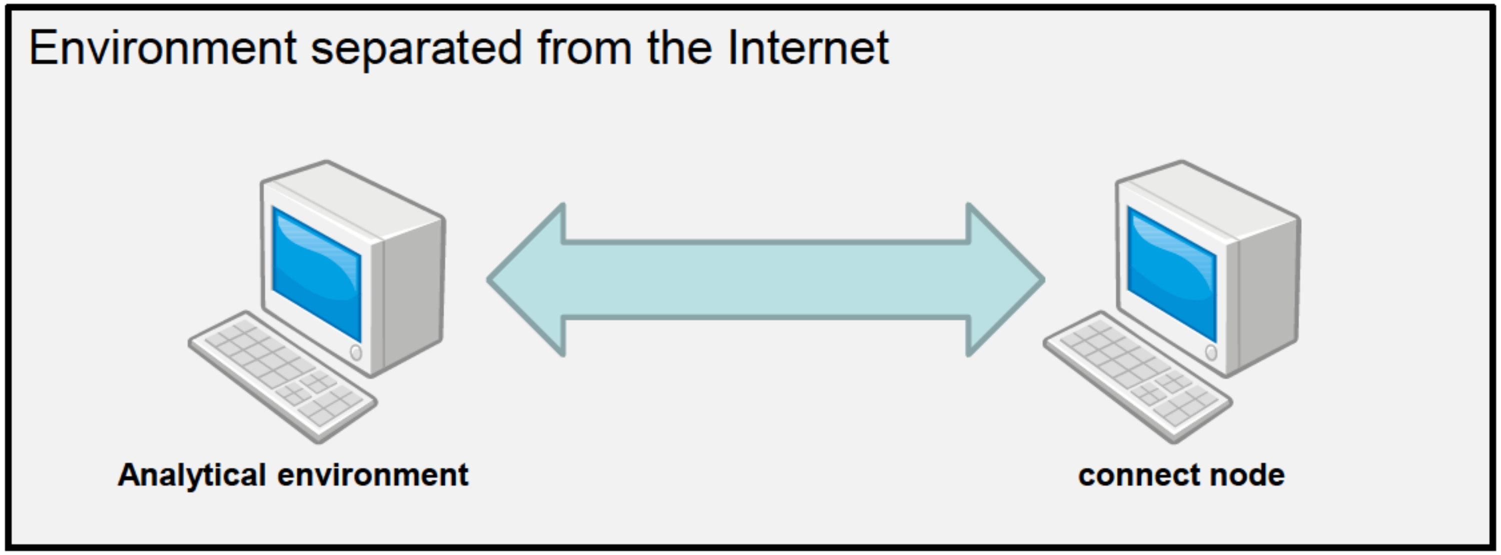
- If all checks was succeeded, the version of connected node was specified.
- If check was failed, checks are done again with other encryption key.

## 2. The approaches to the statical analysis for anonymous P2P filesharing systems



## Building analytical environment

- When the P2P application is connected with a usual network, there are a lot of connections, and the analysis is difficult.
- Therefore, the environment that can be communicated by one-on-one is necessary.



## Anti debugging,  packing

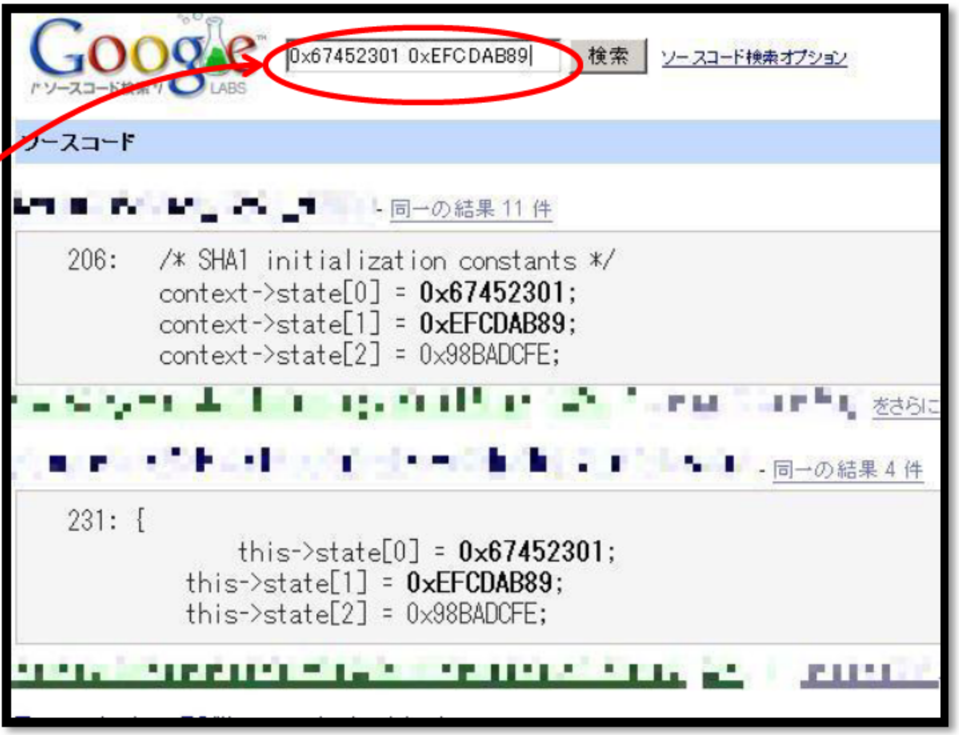
- To improve the anonymity of the P2P software, Anti debugging and packing are given.
- To evade these, it doesn't start by the debugger but it is made to attach by the debugger after the P2P application starts.
- Even if the analysis of initialization flies to some degree, the analysis by this method is possible because it is unquestionable when communication processing is analyzed.

## Analysing network process

- Because the execution file cannot be read with IDA Pro, it is difficult to specify communication processing.
- The location where communication processing is done is specified by setting the breakpoint to API, and tracing the stack.
- Specific in the file access part etc. is also possible by a similar method

## Guessing encryption algorithm

- Even if the assembly code of the cryptographic algorithm is analyzed, specific of the algorithm is difficult.
- Code Search Engine can be used to guess the cryptographic algorithm.



Google ソースコード検索 LABS

0x67452301 0xEFCDAB89 検索 ソースコード検索オプション

ソースコード

[redacted] - 同一の結果 11 件

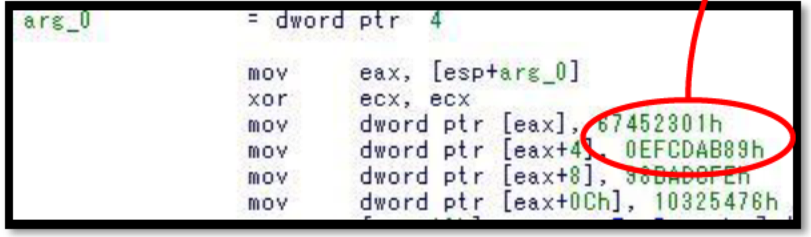
```
206:   /* SHA1 initialization constants */
       context->state[0] = 0x67452301;
       context->state[1] = 0xEFCDAB89;
       context->state[2] = 0x98BADCFE;
```

[redacted] をさらに

[redacted] - 同一の結果 4 件

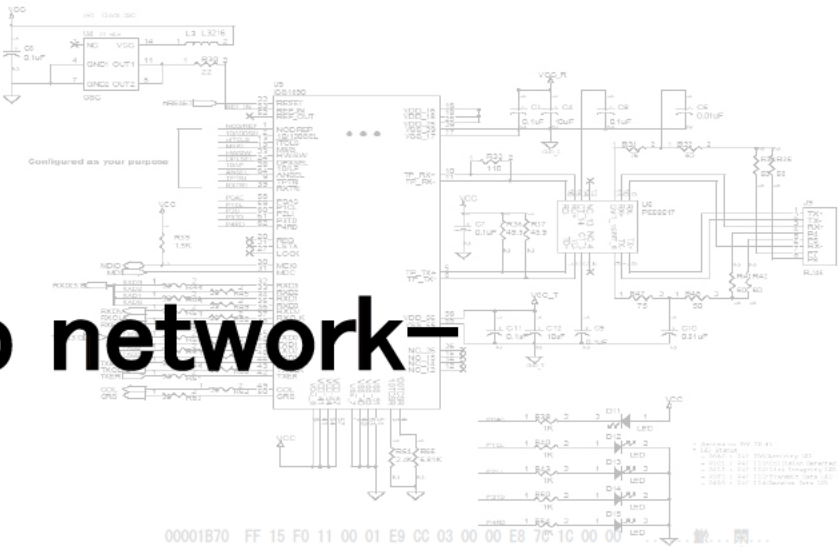
```
231: {
           this->state[0] = 0x67452301;
       this->state[1] = 0xEFCDAB89;
       this->state[2] = 0x98BADCFE;
```

[redacted]



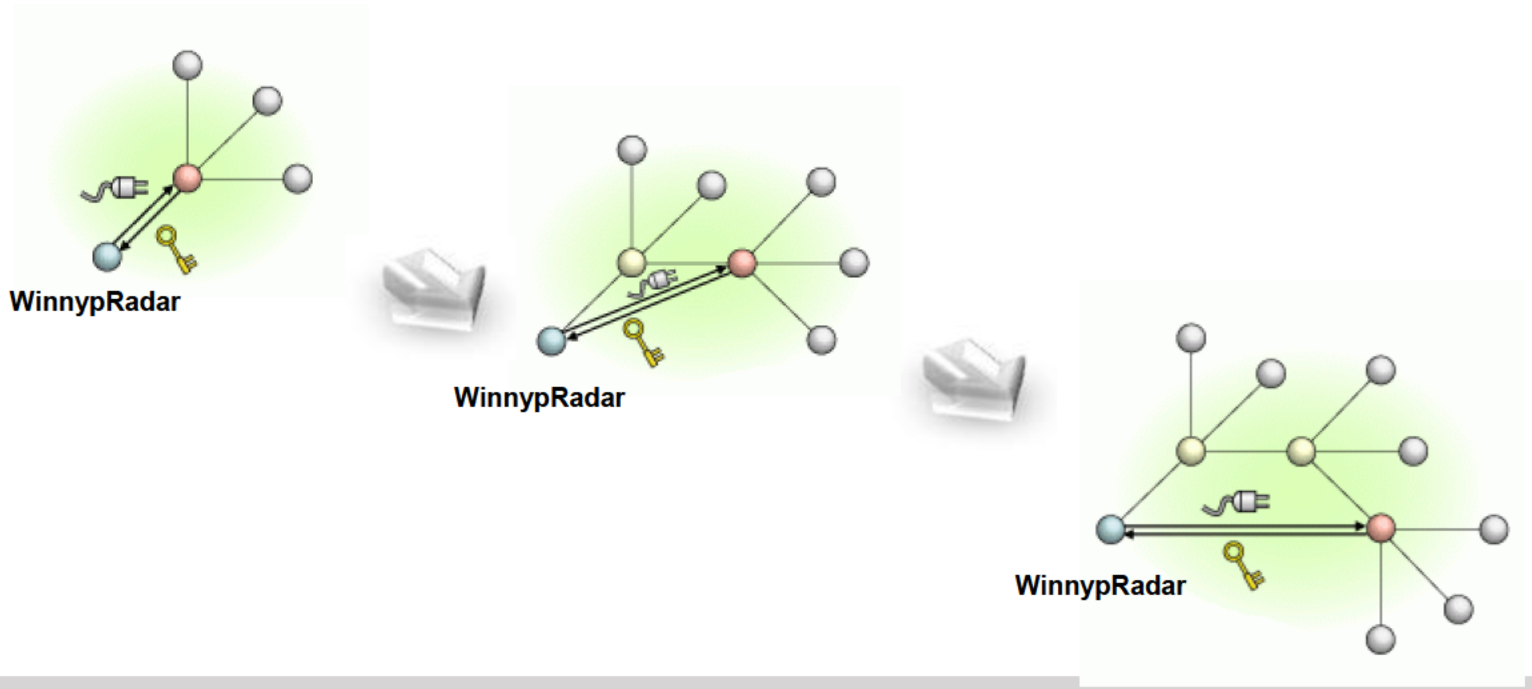
```
arg_0           = dword ptr  4

                mov     eax, [esp+arg_0]
                xor     ecx, ecx
                mov     dword ptr [eax], 67452301h
                mov     dword ptr [eax+4], 0EFCDAB89h
                mov     dword ptr [eax+8], 98BADCFEh
                mov     dword ptr [eax+0Ch], 10325476h
```

# 3. WinnypRadar
## – Crawling Winnyp network–



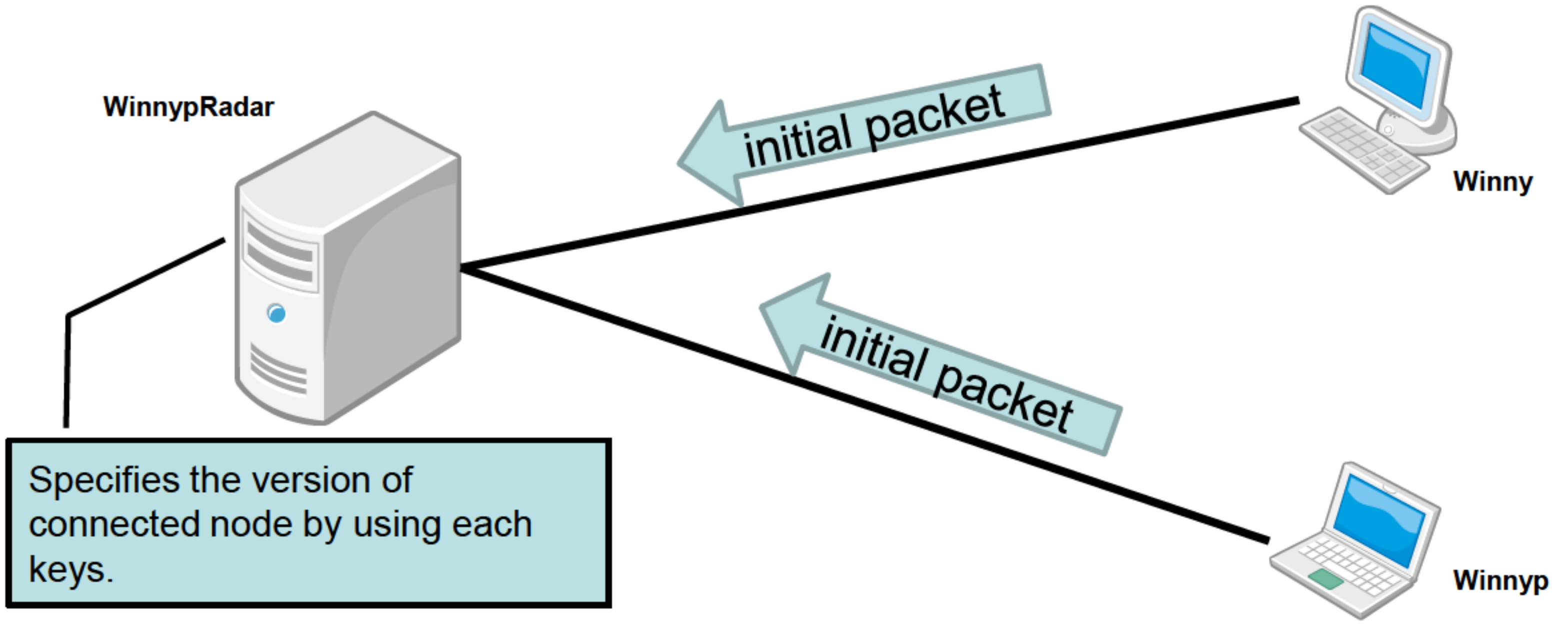
## WinnypRadar

- Connect to Winnyp network as one of the Winnyp node.
- Collects "Key information" from connected node by using Winnyp protocol.
- WinnypRadar can connect to both winny node and winnyp node.



## Distinction between Winny node and Winnyp node

- There is a possibility including the Winny node in the Winnyp network, because Winnyp has compatibility with Winny.
- In WinnypRadar, the version of connected node is specified based on an initial packet of the connected node.



Specifies the version of connected node by using each keys.

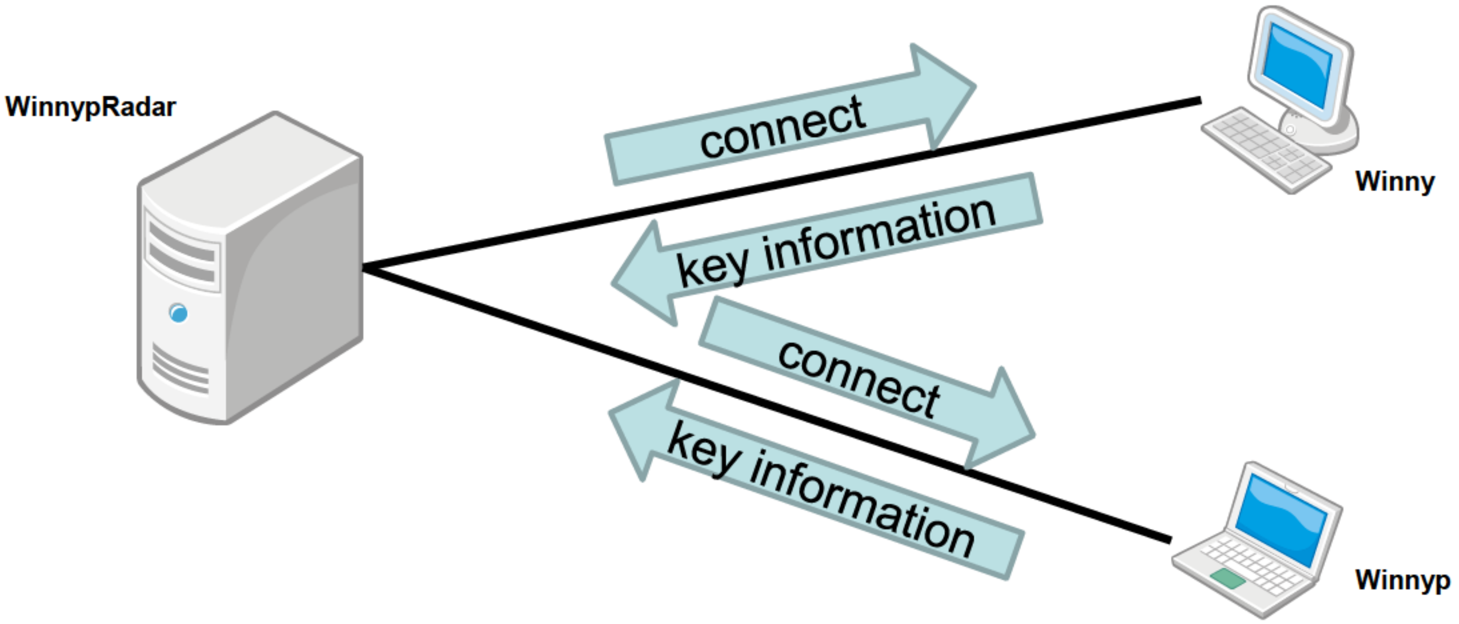
## Collected information

- Collected key information includes file name and IP address.
- What kind of file which IP address has opened to the public can be investigated.

| |
|---|
| **IP address** |
| **Port** |
| **File size** |
| **File time stamp** |
| **File name** |
| **Hash** |
| **…** |

## Crawling

- IP address in the collected key information is used to the new connection destination.
- Connects at the new connection destination, and key information is acquired.



## Result

- Proportion of Winnyp node in Winny network
  - Be judged whether connected node is Winnyp when WinnypRadar connects it, and records.
  - The ratio of Winnyp is calculated from the number of connected all nodes.
- Result of measurement during a day by using WinnypRadar

| | |
|---|---|
| **Node count** | **198,000 node（exclude Port0 setting）** |
| Proportion of Winnyp node | 8%（16,000 node） |

※From analysis result by CROSSWARP, Inc
http://www.scat.or.jp/stnf/contents/p2p/p2p080910_4.pdf

## Conslusion

- I analyzed the encryption key generation algorithm and packet processing of Winnyp v2.1b7.28
- In Winnyp, the encryption key generation algorithm is more complex than Winny.
- Develop Winnyp network crawler “WinnypRadar” based on the analytical result.
- The investigation concerning the Winnyp node that was not able to be detected up to now by the use of WinnypRadar became possible.

## Further tasks

- This time, because an enough node investigation period was not able to be taken, it is necessary to investigate the node for the long term.

- It seems that a more accurate number of Winnyp nodes can be measured by investigating the node that connects it only with Winnyp though the Winnyp node that was able to be connected with the Winny network was investigated in this investigation.

## Thank you!

Fourteenforty Research Institute, Inc.
http://www.fourteenforty.jp

Senior Software Engineer
Toshiaki Ishiyama
ishiyama@fourteenforty.jp