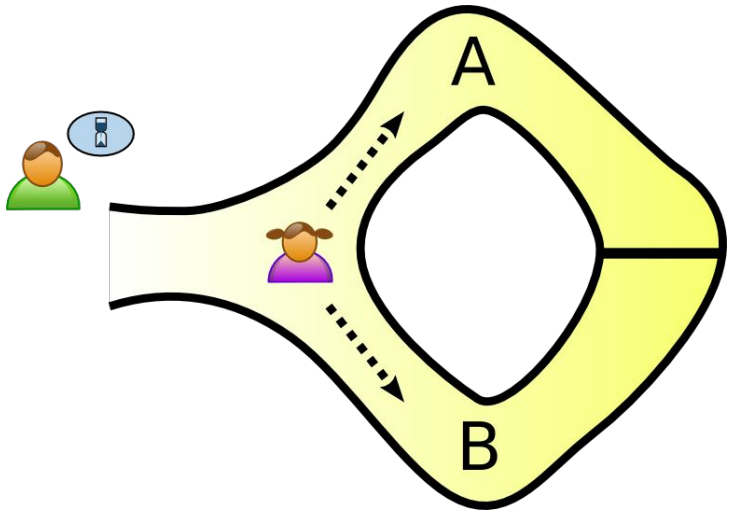
UCI University of California, Irvine

MITRE eCTF 2023-24

Reverse Engineering w/ Ghidra

Jinyao Xu (jinyaox@uci.edu)

Zero Knowledge Proof



Q: **How to prove you can solve a sudoku without giving out the answer?**

- **Garble the solution and let other verify**

Q: What about a k-variable Polynomial? Hummmm…

How Ghidra or IDA works?

Disassembly is NOT Magic

1. Linear Sweep:

   Read the first N-bytes until you get a correct opcode (e.g., 05 14 00 00 00 decompiles to add eax,0x14). Disassemble the next opcode.

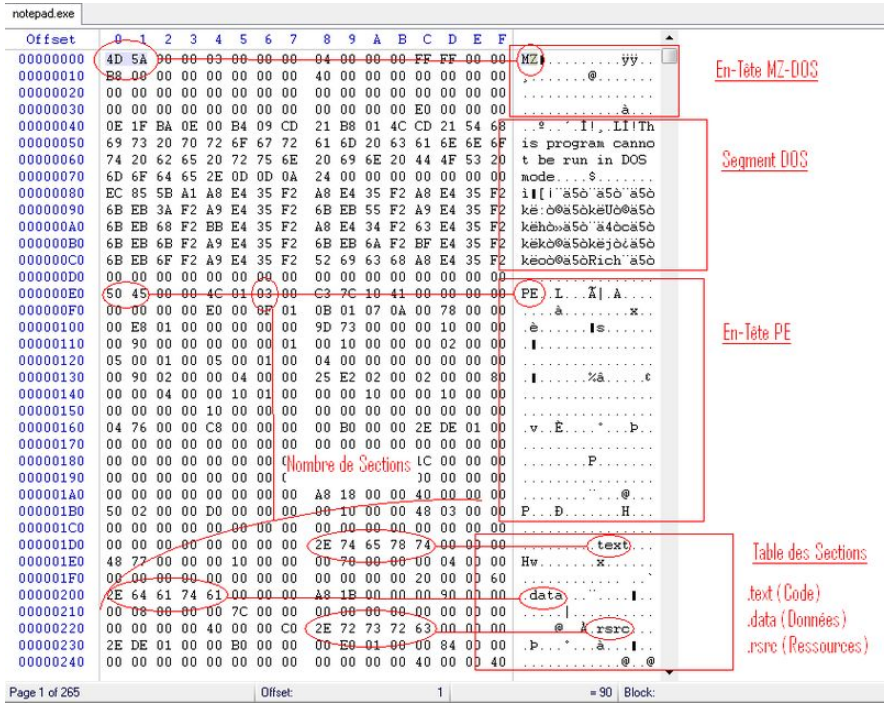
2. Recursive Traversal:

   Read the first N-bytes until you get a correct opcode. Proceed until you disassemble any sort of jump, store your current position, follow the jump and proceed as described. Stop disassembly when you get an invalid opcode and resume at the previously stored position.

How Ghidra or IDA works?

```
8b 4c 24 04 8b c1 99 33 c2 2b c2 83 e0 01 33 c2
2b c2 8d 44 49 01 74 07 8d 04 8d fd ff ff ff c3
```

```
 0:   8b 4c 24 04             mov    ecx, [esp+4]
 4:   8b c1                   mov    eax, ecx
 6:   99                      cdq
 7:   33 c2                   xor    eax, edx
 9:   2b c2                   sub    eax, edx
 b:   83 e0 01                and    eax, 1
 e:   33 c2                   xor    eax, edx
10:   2b c2                   sub    eax, edx
12:   8d 44 49 01             lea    eax, [ecx+ecx*2+1]
16:   74 07                   je     01fh
18:   8d 04 8d fd ff ff ff    lea    eax, [ecx*4-3]
1f:   c3                      ret
```



Plain text = P

Encrypted text = C

Jet: $[k_{p_1}, k_{p_2}, k_{p_3}]$

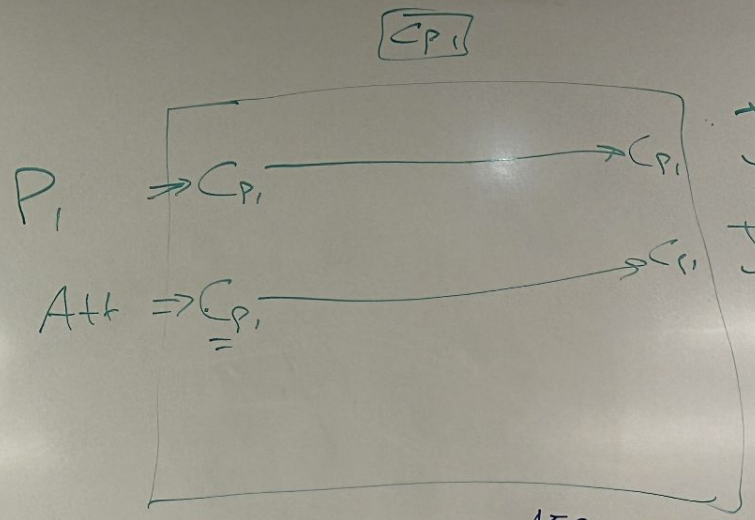
$P_1 : k_{p_1}$

$P_2 : k_{p_2}$

$P_3 : k_{p_3}$

$E(P \cdot k_{p_1} \cdot N) = C_{p_1}$

$D(C \cdot k_{p_1}) = P_{p_1}$



AES: 2 ms.

1. Goal 1 : Ciphertext indistinguishability.

CSV file.

- py. load procs.
- py. generate nonce.
- py. validate
- py. send out.

Interval: 0.1 sec.

100 ms.

JET

Pilot

- EEPROM STORE info. AES private key
- [Recieve Encrypt Send.] = Button push

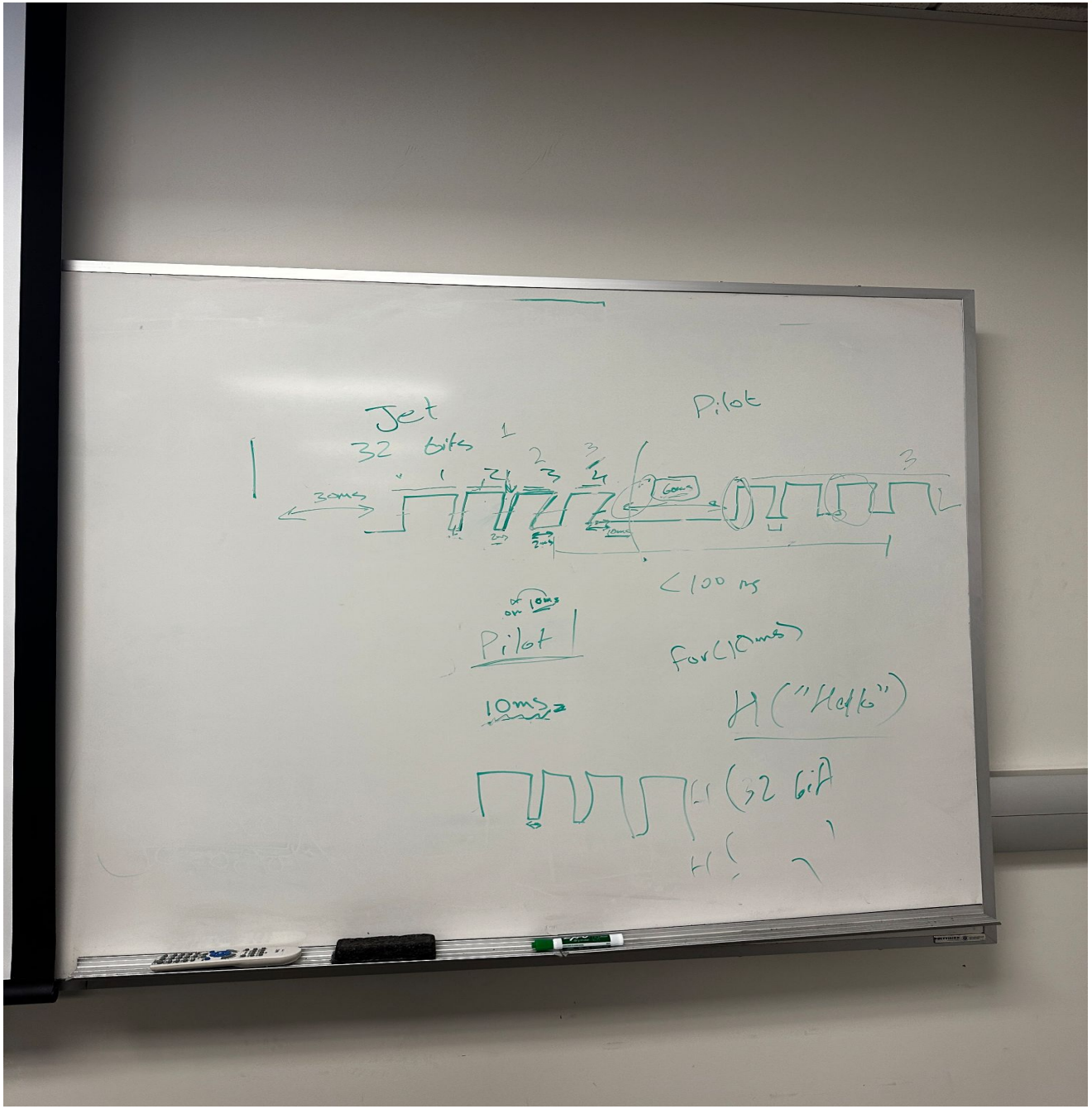
K_{P1} → 3 →

← C_{P1}

$D_{K1}(C_{P1}) = 3$

K_{P1} UART Rx AES Lib UART Tx

$E_{K1}(3) = C_{P1}$


Jet
32 bits
Pilot
30ms
< 100 ms
Pilot
10ms
for 10ms
H("Hello")