

# CCBHash (Compound Code Block Hash) para Análisis de Malware

**Pablo Pérez, José Antonio Onieva**

*Network, Information and Computer Security (NICS) Lab  
Universidad de Málaga*

**Gerardo Fernández**

*VirusTotal  
Málaga*

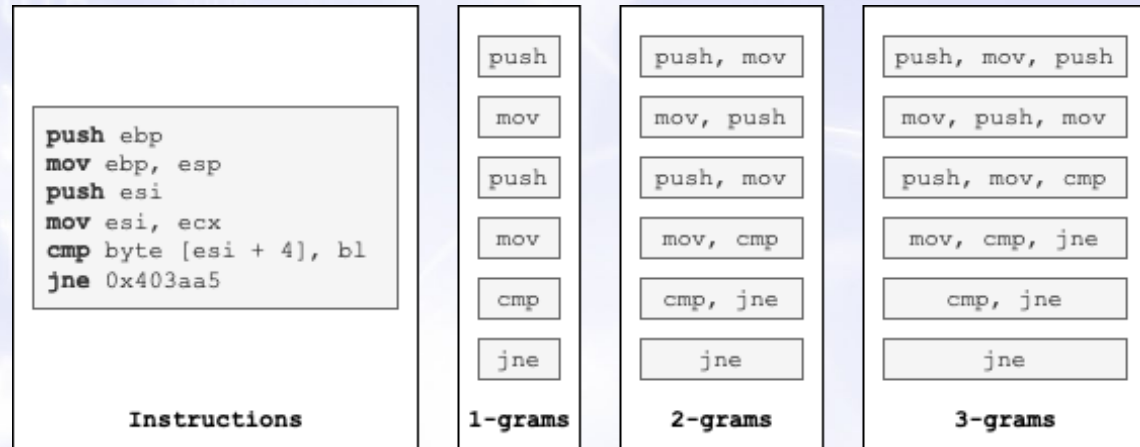
*Santander, Octubre 2022*

# Buscadores de similitud

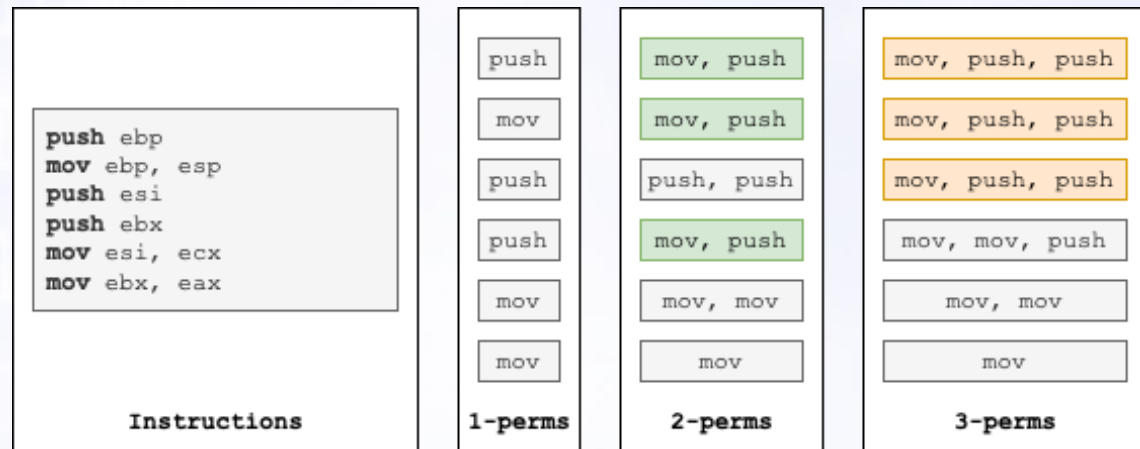
- Similitudes entre numerosos ficheros:
  - Estrategias basadas en n-grams
  - Estrategias basadas en grafos
  - Estrategias basadas en Fuzzy Hashing
  - ...
- Similitudes entre funciones de dos ficheros:
  - BinDiff
  - Diaphora
- Similitudes entre numerosas funciones:
  - ?

# Similitudes entre numerosos ficheros

- N-grams



- N-perms

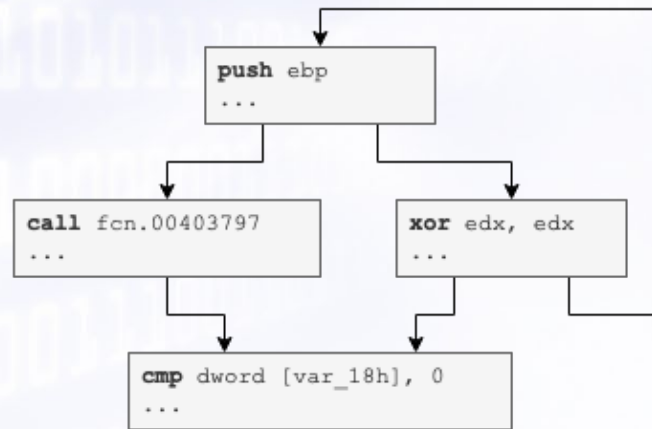


# Similitudes entre numerosos ficheros

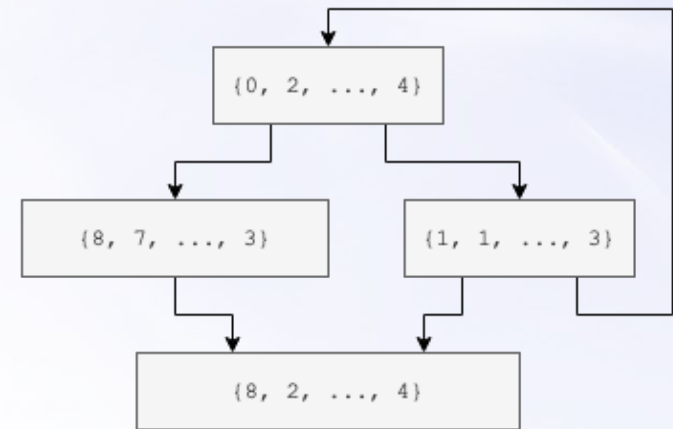
- Grafos
  - Callgraph
  - CFG (Control Flow Graph)
  - ACFG (Attributed CFG)

```
push ebp
mov ebp, esp
...
cmp dword [var_18h], 0
je 0x4072a7
...
call fcn.00403797
cmp dword [var_14h], eax
jb 0x403b93
...
xor edx, edx
add edi, ecx
...
```

Instructions



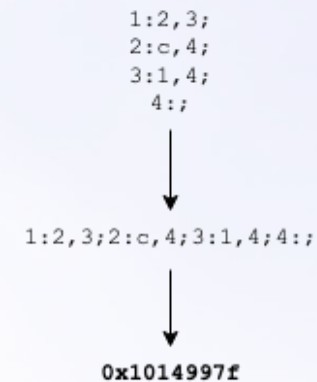
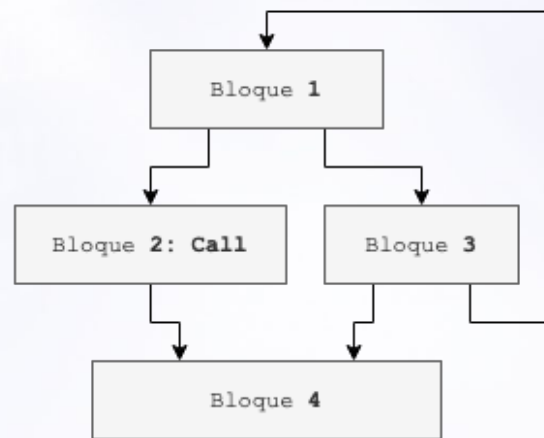
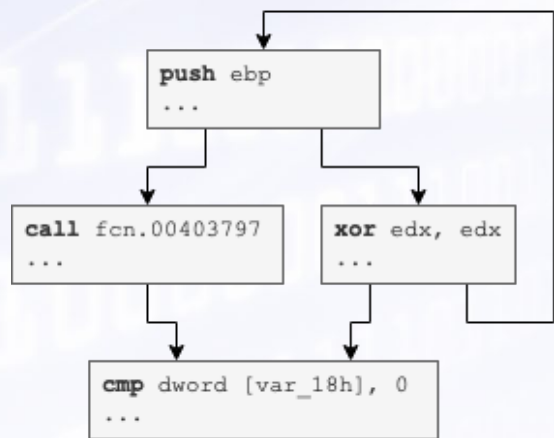
Control Flow Graph



Attributed Control Flow Graph

# Similitudes entre numerosos ficheros

- Fuzzy Hashing
  - BBH (Block Based Hashing)
  - CTPH (Context Triggered Piecewise Hashing)
  - SIF (Statistically Improbable Features)
  - BBR (Block Based Rebuilding)
  - LSH (Locality Sensitive Hashing)
- Machoc y Machoke





# Similitudes entre funciones de dos ficheros

- BinDiff y Diaphora

Workspace: sub\_140008950 vs sub\_00408572 x | sub\_140001D18 vs sub\_00403586 x | sub\_1400036E0 vs sub\_00404CC0 x | sub\_1400050FC vs sub\_00407137 x

Overview: Basic Blocks 100.0% | Jumps 100.0% | Instructions -380.0% | Similarity 0.48

336 / 336 Matched Functions

| Similarity | Confidence | Address         | Primary Name  | Type   | Address  | Secondary Name | Type   | Basic Blocks | Jumps |
|------------|------------|-----------------|---------------|--------|----------|----------------|--------|--------------|-------|
| 0.91       | 0.94       | 000000014000... | sub_14000708C | Normal | 00405558 | sub_00405558   | Normal | 0 3 0 0      | 3 0   |
| 0.91       | 0.94       | 000000014000... | sub_140003E60 | Normal | 0040557C | sub_0040557C   | Normal | 0 3 0 0      | 3 0   |
| 0.90       | 0.94       | 000000014000... | sub_140003FF0 | Normal | 004056CC | sub_004056CC   | Normal | 0 3 0 0      | 3 0   |
| 0.90       | 0.92       | 000000014000... | sub_14000A460 | Normal | 004043B0 | sub_004043B0   | Normal | 0 11 0 0     | 13 0  |
| 0.90       | 0.92       | 000000014000... | sub_140008F2C | Normal | 004068E8 | sub_004068E8   | Normal | 0 5 0 0      | 7 0   |
| 0.90       | 0.94       | 000000014000... | sub_14000765C | Normal | 00409651 | sub_00409651   | Normal | 0 3 0 0      | 3 0   |
| 0.90       | 0.93       | 000000014000... | sub_140006854 | Normal | 00408A00 | sub_00408A00   | Normal | 0 4 0 0      | 5 0   |
| 0.90       | 0.94       | 000000014000... | sub_1400084CC | Normal | 00405AD5 | sub_00405AD5   | Normal | 0 5 0 0      | 6 0   |
| 0.90       | 0.93       | 000000014000... | sub_1400087F4 | Normal | 0040569F | sub_0040569F   | Normal | 0 4 0 0      | 4 0   |
| 0.90       | 0.98       | 000000014000... | sub_1400036E0 | Normal | 00404CC0 | sub_00404CC0   | Normal | 0 6 1 1      | 8 2   |
| 0.90       | 0.93       | 000000014000... | sub_140004320 | Normal | 00405B55 | sub_00405B55   | Normal | 0 4 0 0      | 4 0   |
| 0.89       | 0.91       | 000000014000... | sub_14000A324 | Normal | 00409220 | sub_00409220   | Normal | 0 3 0 0      | 3 0   |
| 0.89       | 0.92       | 000000014000... | sub_140004590 | Normal | 00405EB0 | sub_00405EB0   | Normal | 0 6 0 0      | 7 0   |
| 0.89       | 0.92       | 000000014000... | sub_1400043A0 | Normal | 00405D57 | sub_00405D57   | Normal | 0 3 0 0      | 3 0   |
| 0.88       | 0.92       | 000000014000... | sub_1400098C8 | Normal | 00406742 | sub_00406742   | Normal | 0 3 0 0      | 3 0   |
| 0.88       | 0.90       | 000000014000... | sub_140009B90 | Normal | 00408336 | sub_00408336   | Normal | 0 8 0 0      | 11 0  |
| 0.88       | 0.91       | 000000014000... | sub_1400034D0 | Normal | 00404A88 | sub_00404A88   | Normal | 0 4 0 0      | 4 0   |
| 0.88       | 0.90       | 000000014000... | sub_140004198 | Normal | 00405841 | sub_00405841   | Normal | 0 13 0 0     | 17 0  |
| 0.88       | 0.91       | 000000014000... | sub_140005A60 | Normal | 00407968 | sub_00407968   | Normal | 1 23 0 2     | 31 0  |
| 0.88       | 0.92       | 000000014000... | sub_140008D20 | Normal | 00409195 | sub_00409195   | Normal | 0 3 0 0      | 3 0   |
| 0.87       | 0.90       | 000000014000... | sub_140006DF8 | Normal | 00408FD1 | sub_00408FD1   | Normal | 0 4 0 0      | 4 0   |
| 0.87       | 0.89       | 000000014000... | sub_140003790 | Normal | 00404DD0 | sub_00404DD0   | Normal | 0 5 0 0      | 6 0   |
| 0.87       | 0.91       | 000000014000... | sub_140003B00 | Normal | 004086B0 | sub_004086B0   | Normal | 0 6 0 0      | 8 0   |
| 0.87       | 0.96       | 000000014000... | sub_1400050FC | Normal | 00407137 | sub_00407137   | Normal | 0 11 2 2     | 14 4  |
| 0.86       | 0.89       | 000000014000... | sub_140009EAC | Normal | 00408645 | sub_00408645   | Normal | 0 5 0 0      | 6 0   |
| 0.86       | 0.95       | 000000014000... | sub_140006C04 | Normal | 00408CE3 | sub_00408CE3   | Normal | 1 13 1 4     | 16 4  |

# Similitudes entre funciones de dos ficheros

- BinDiff y Diaphora

The screenshot displays the BinDiff interface comparing two assembly functions. The left pane shows the 'primary' function, `sub_1400036E0`, and the right pane shows the 'secondary' function, `sub_00404CC0`. Both functions are shown as control flow graphs with nodes containing assembly instructions. The instructions in both functions are nearly identical, demonstrating high similarity between the two binaries.

**Primary Function: `sub_1400036E0`**

```
0000001400036E0 sub_1400036E0
0000001400036E0 movxd r8, b4 ds:[rcx+0x3C]

0000001400036E4 xor b4 r9d, b4 r9d
0000001400036E7 add r8, rcx

0000001400036EA mov r10, rdx

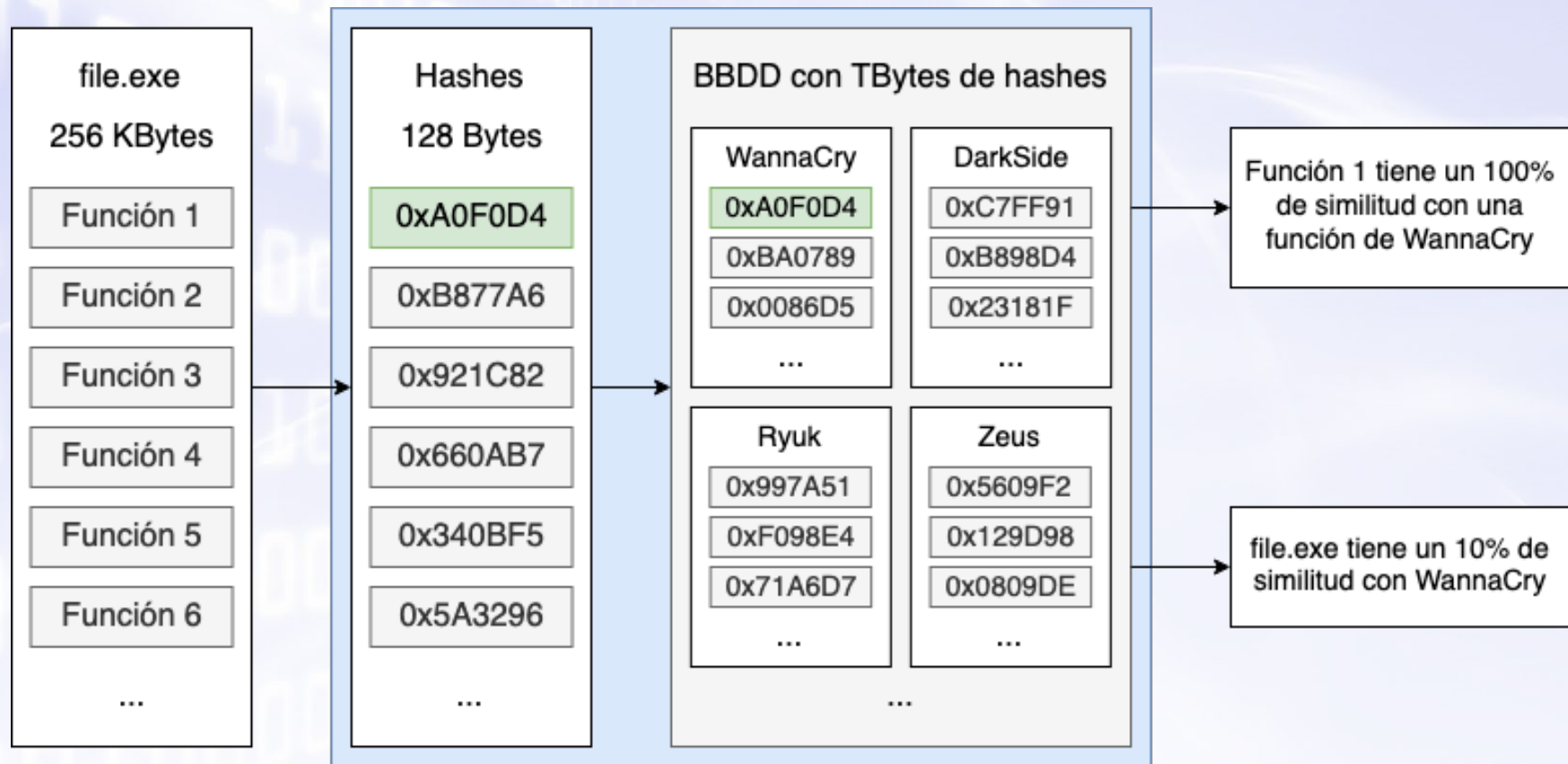
0000001400036ED movzx b4 eax, b2 ds:[r9+0x14]
0000001400036F2 movzx b4 r11d, b2 ds:[r8+6]
0000001400036F7 add rax, b1 0x10
0000001400036FB add rax, r8
0000001400036FE test b4 r11d, b4 r11d
000000140003701 jz 0x140003721
```

**Secondary Function: `sub_00404CC0`**

```
00404CC0 sub_00404CC0
00404CC0 mov edi, edi
00404CC2 push ebp
00404CC3 mov ebp, esp
00404CC5 mov eax, ss:[ebp+arg_0]
00404CC8 xor edx, edx

00404CCA push ebx
00404CCB push esi
00404CCD push edi
00404CCD mov ecx, ds:[eax+0x3C]
00404CD2 add ecx, eax
00404CD2 movzx eax, b2 ds:[ecx+0x14]
00404CD6 movzx ebx, b2 ds:[ecx+6]
00404CDA add eax, b1 0x10
00404CDD add ecx, eax
00404CDF test ebx, ebx
00404CE1 jz 0x404CFE
```

# CCBHash (Compound Code Block Hash)





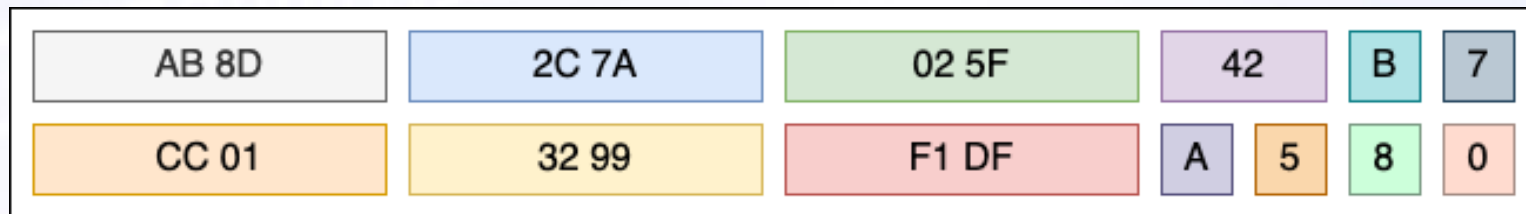
# Diseño de CCBHash

- Nombre de la función (2 bytes)
- CFG de la función (2 bytes)
- Callgraph de la función (2 bytes)
- Tipos de opcodes (2 bytes)
- Tipos de argumentos (2 bytes)
- Tipos de variables locales (2 bytes)
- Número de instrucciones (1 byte)
- Número de bloques (4 bits)
- Número de entradas (4 bits)
- Número de salidas (4 bits)
- Número de args + vars (4 bits)
- Complejidad ciclomática (4 bits)
- Tamaño de la pila (4 bits)

Se almacenan hashes (usando blake2b).

Se almacenan valores numéricos.

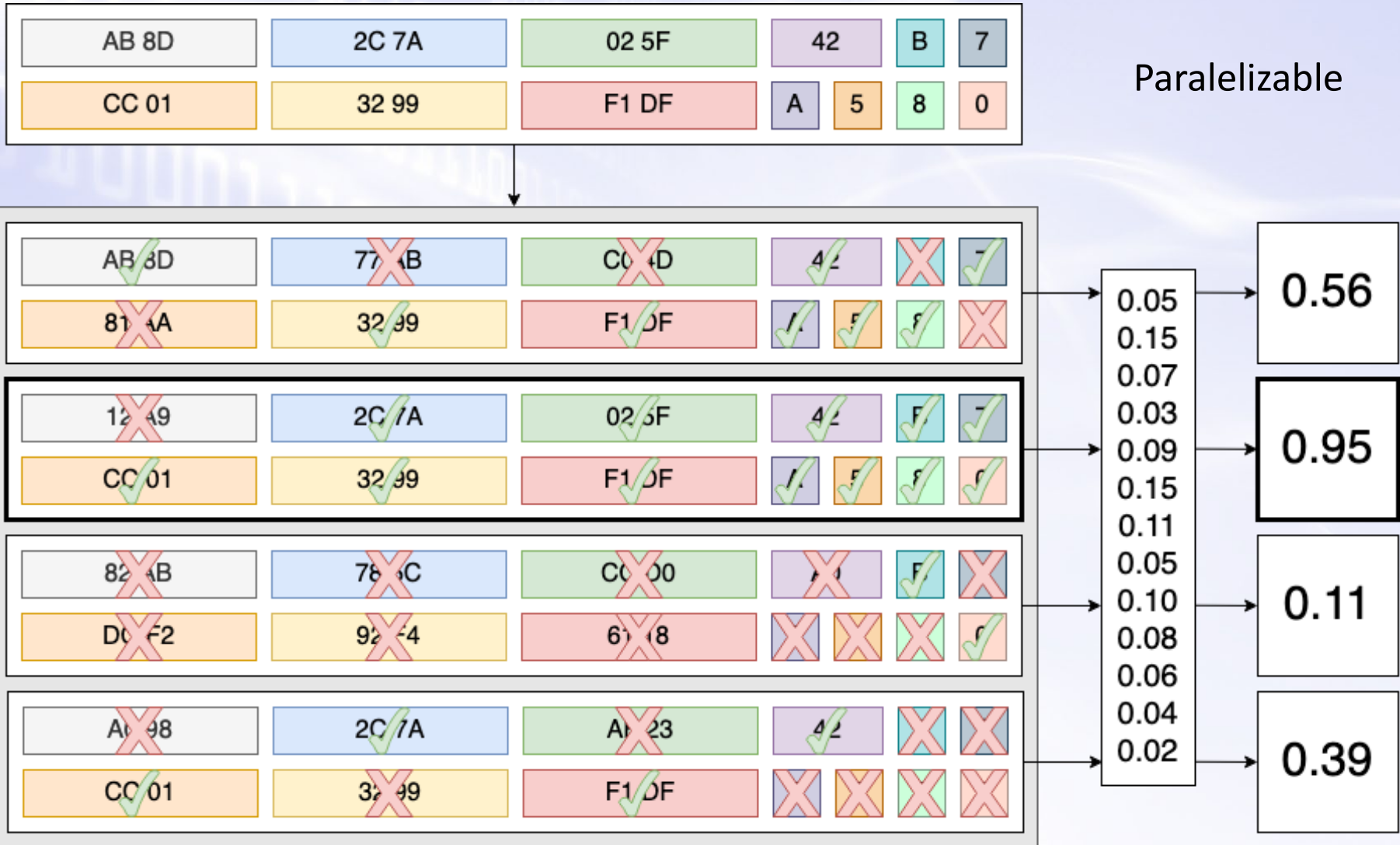
16 bytes



# Diseño de CCBHash

| Atributo                                   | Puntuación |
|--|------------|
| Tipos de opcodes                           | 0.25       |
| CFG de la función                          | 0.16       |
| Complejidad ciclomática                    | 0.15       |
| Número de salidas (outdegree)              | 0.09       |
| Número de instrucciones                    | 0.07       |
| Callgraph de la función                    | 0.06       |
| Tipo de argumentos de la función           | 0.04       |
| Tipo de variables locales                  | 0.04       |
| Cantidad de argumentos y variables locales | 0.04       |
| Número de bloques                          | 0.04       |
| Número de entradas (indegree)              | 0.03       |
| Tamaño de la pila                          | 0.02       |
| Nombre de la función                       | 0.01       |

# Diseño de CCBHash

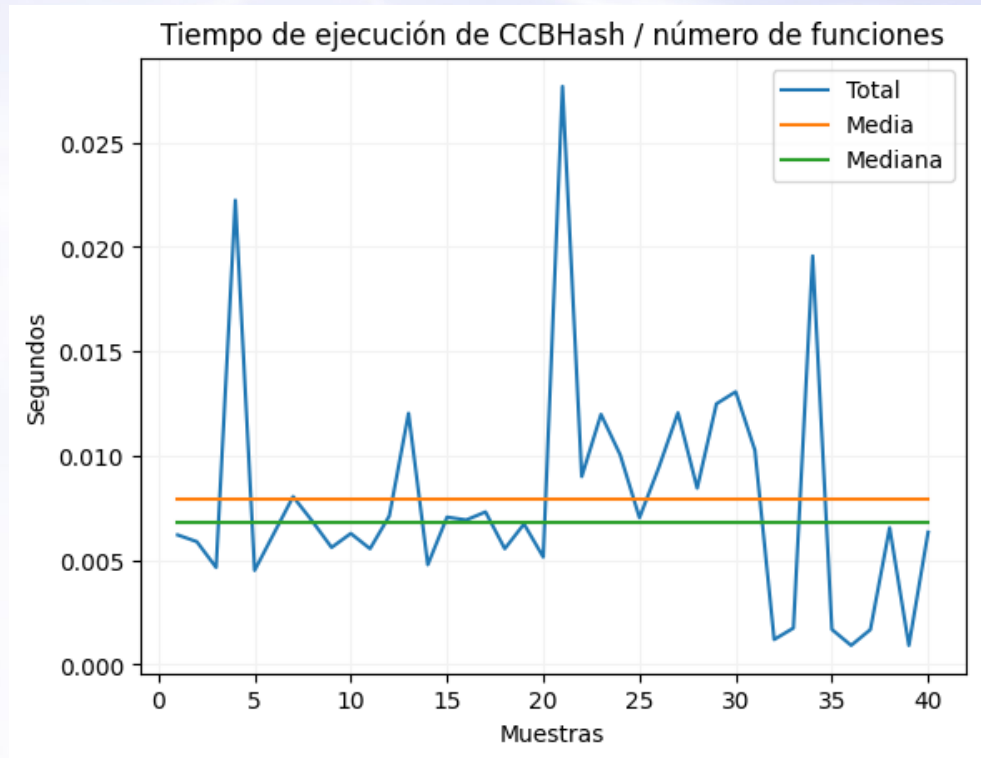
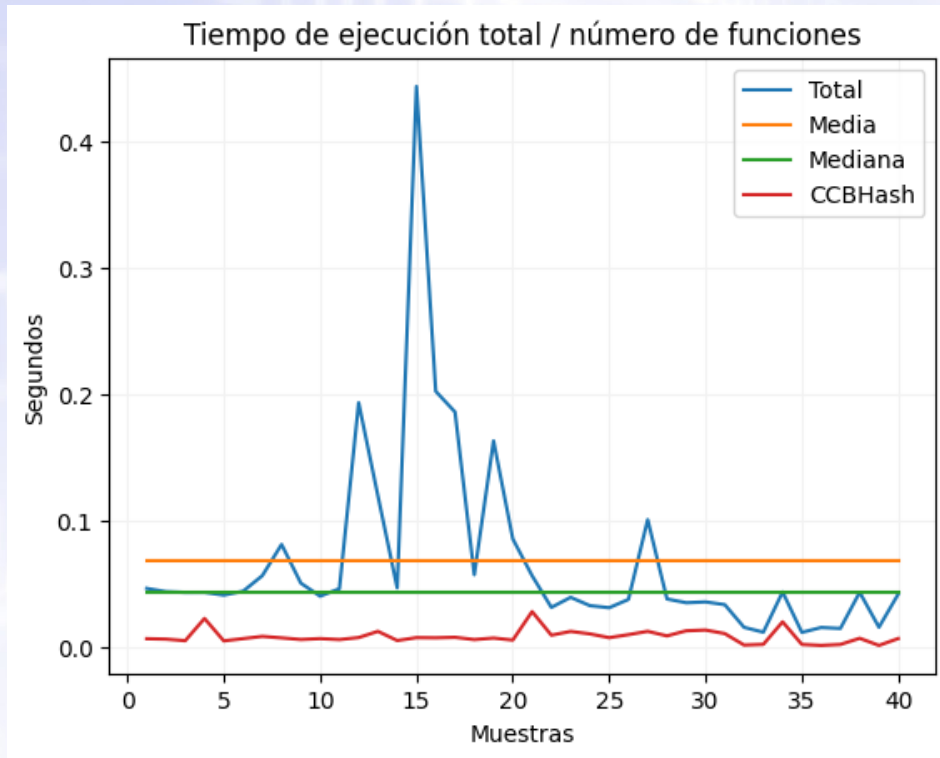


# Implementación de CCBHash



# Análisis de resultados

- Análisis de tiempos:





# Análisis de resultados

- Análisis de precisión:

| Familia de malware | Número medio de ficheros con dicha similitud |        |        |        |        |        |
|--------------------|--|--------|--------|--------|--------|--------|
|                    | CCBHash                                      |        |        | ssdeep |        |        |
|                    | > 90 %                                       | > 75 % | > 50 % | > 90 % | > 75 % | > 50 % |
| WannaCry           | 4.5  | 0      | 0      | 0.4    | 1.1    | 2.3    |
| DarkSide           | 2  | 2      | 2.3    | 0.2    | 0.2    | 0.7    |
| Ryuk               | 6  | 0      | 2*     | 0      | 0      | 0      |
| Zeus               | 3.3  | 0      | 0      | 2.2    | 0      | 0      |
| Total              | 4  | 0.5    | 1.1    | 0.7    | 0.3    | 0.8    |

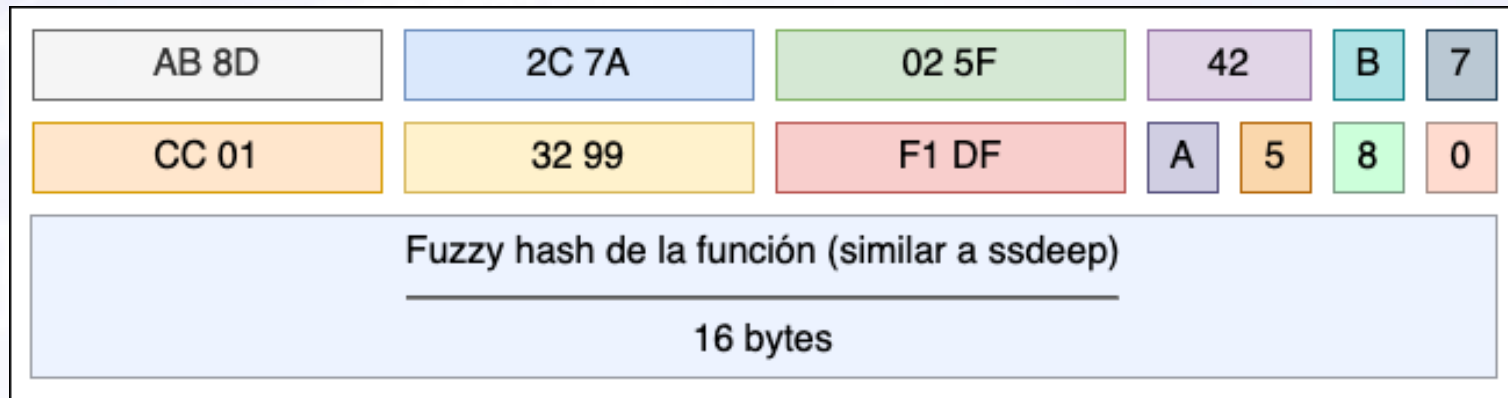
| Familia de malware | Número medio de ficheros con dicha similitud |        |        |      |      |       |
|--------------------|--|--------|--------|------|------|-------|
|                    | CCBHash                                      |        |        | TLSH |      |       |
|                    | > 90 %                                       | > 75 % | > 50 % | < 10 | < 50 | < 100 |
| WannaCry           | 4.5  | 0      | 0      | 0    | 0    | 0.6   |
| DarkSide           | 2  | 2      | 2.3    | 0.2  | 0.8  | 0.5   |
| Ryuk               | 6  | 0      | 2*     | 0    | 0.2  | 1     |
| Zeus               | 3.3  | 0      | 0      | 2    | 0    | 0     |
| Total              | 4  | 0.5    | 1.1    | 0.6  | 0.3  | 0.5   |

# Conclusiones

- Solución a un problema actual y complejo
- Comparación rápida de funciones
- Comparación rápida de ejecutables
- Mejora a fuzzy hashes como ssdeep o TLSH

# Trabajo futuro

- Mejora de atributos
- Optimización de filtros de funciones
- Plugin para IDA Pro
- Estudio de colisiones para atributos con hashes
- Análisis con dataset más amplio
- Versión extendida de 32 bytes



# CCBHash (Compound Code Block Hash) para Análisis de Malware

**Pablo Pérez, José Antonio Onieva**

*Network, Information and Computer Security (NICS) Lab  
Universidad de Málaga*

**Gerardo Fernández**

*VirusTotal  
Málaga*

*Santander, Octubre 2022*