

OBFUSKÁCIA S VYUŽITÍM LLVM

Roman Oravec
Europen 2022



0 čom to bude?

- 1 Obfuskačné transformácie
- 2 LLVM
- 3 Implementácie

OBFUSKÁCIA

Transformuje
program

Cieľom je
stíhať analýzu

Funkcionalita
musí byť
zachovaná

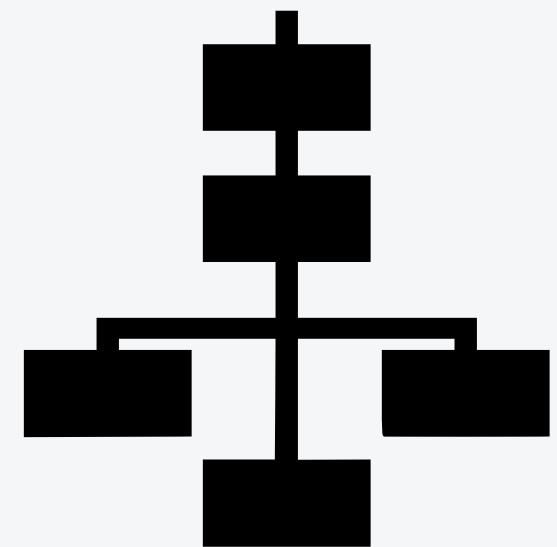
```

#\
define C(c) /***)#c
/*size=3173*/#include<stdio.h>
/*crc=b7f9ecff.*#/include<stdlib.h>
/*Mile/Adele_von_Aschar*/#include<time.h>
typedef/**/int(I);I/*:3*/d,i,j,a,b,l,u[16],v
[18],w[36],x,y,z,k;char*P="\n\40(),",*p,*q,*t[18],m[4];
void/**/O(char*q){for(;*q;q++)*q>32?z=111-*q?z=(z+*q)%185,(k?
k--:(y=z%37,(x=z/37%7)?printf(*t,t[x],y?w[y-1]:95):y>14&&y<33?x
=y>15,printf(t[15+x],x?2<<y%16:l,x?(1<<y%16)-1:1):puts(t[y%28])))
,0:z+82:0;}void/**/Q(I(p),I*q){for(x=0;x<p;x++){q[x]=x;}for(--p
=q[x=rand()%~p],q[x]=q[p];}char/**/n[999]=C(Average?!nQVQd%R>Rd%
>1;q[p]=y)y
R% %RNIPRfi#VQ}R;TtuodtsRUD%RUD%RUOSetirwf!RnruterR{RTSniamRtniQ>h.oidts<edulc
ni #V>rebmun<=NIPD-RhtiwRelipmocResaelPRrorre#QNIPRfednfi#V__ELIF__R_
Re nifed#V~-VU0V;}V{R= R][ORrahcRdengisnuRtsnocRcitatsVesle#Vfidne#V53556
. .1RfoRegnarRehtRniRre getniRnaRsiR]NIP[R erehwQQc.tuptuoR>Rtxt.tupniR
< R]NIP[R:egasuV_Redulcn i#VfednfiVfednuVenife dVfedfiVQc%Rs%#V);I/**/main(
I( f),char**e){if(f){for(i= time(NULL),p=n,q= n+998,x=18;x;p++){*p>32&&!(
*--q=*p>80&&*p<87?P[*p- 81]:* p)?t [( -- x)]=q+1:q;}if(f-2||(d=atoi
(e[1]))<1||65536<d){;O(" \"); goto O;}srand(i);Q(16,u);i=0;Q(
36,w);for(;i<36; i++){w[i] +=w [i]<26 ? 97:39; }O(C(ouoo9oBotoo%)#
ox^#oy_#ozoou#o{ a#o|b#o}c#
2j#oo3k#oo4l#o p));for(j =8;EOF -(i= getchar());l+=1){a=1+
rand()%16;for(b =0;b<a|i-
32,d= (d/ 2|x<<15)&65535;
a++ ){if( (b&(1<<(i=v[a] )))* main (0,e);b++)x=d^d/4^d/8^d/
!) ); }O(C(oqovoo97o /n!);i= b|= !l<<17;Q(18,v);for(a=0;a<18;
]= 75 +i++;O(C(oA!oro oqoo9) ) ;m=75+i,O(m),j=i<17&&j<i?i:j; }O(C(
!W !W #2 | !V!V#1{ !U!U#0z! T!T#/y!S!S#.x!R!R#-w!Q!Q#ooAv!P!P#+o#!O!O##t!N!
N# oo >s!M!M#oo=r!L!L#oo<q!K!K# &pIo@:;= oUm#oo98m##oo9=8m#oo9oUm###oo9;=8m#o
o9 oUm##oo9=oUm#oo98m#### o09] #o1:^#o2;_#o3<o ou#o4=a#o5>b#o6?c#o
7@d#o8A e#o 9B f#o:Cg#o; D h#o<Ei #o=Fj#o> Gk#o?Hl#oo9os#####;
));d=0 ;} O: for(x=y=0;x<8;++x)y |=
x<< 1<<
/* y

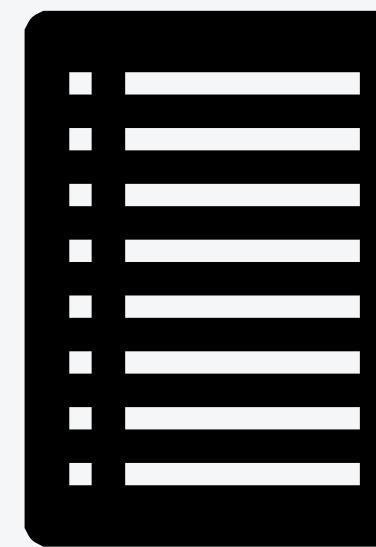
```

Čo obfuskovať?

Riadiace
štruktúry



Dáta



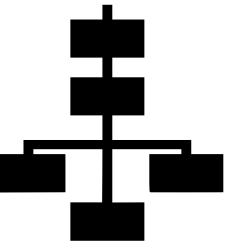
Kto je "utočník"?

Reverzní
inžinieri



Nástroje

- Triton
- Angr
- Z3
- . . .



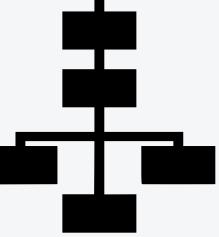
Zámena inštrukcií

$$x + y \rightarrow -(-x + (-y))$$

$$x - y \rightarrow x + (-y)$$

$$x \vee y \rightarrow (x \wedge y) \vee (x \oplus y)$$

$$x \oplus y \rightarrow (\neg x \wedge y) \vee (x \wedge \neg y)$$



```
int do_stuff(int r){  
    int a = 4;  
    int b = 2;  
    r /= a ^ b;  
    return r;  
}
```

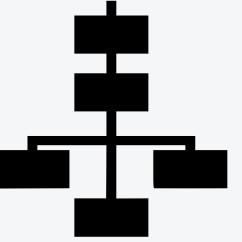


```
mov    DWORD PTR [rbp-0x8],0x4  
mov    DWORD PTR [rbp-0xc],0x2  
mov    eax,DWORD PTR [rbp-0x8]  
xor    eax,DWORD PTR [rbp-0xc]  
mov    ecx,DWORD PTR [rbp-0x4]  
mov    DWORD PTR [rbp-0x10],eax  
mov    eax,ecx  
cdq  
mov    ecx,DWORD PTR [rbp-0x10]  
idiv  ecx
```



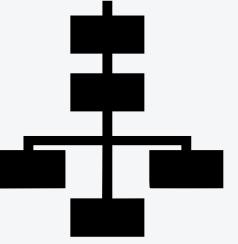
```
mov    DWORD PTR [rbp-0x8],0x4  
mov    DWORD PTR [rbp-0xc],0x2  
mov    eax,DWORD PTR [rbp-0x8]  
mov    ecx,DWORD PTR [rbp-0xc]  
mov    edx, eax  
xor    edx,0xffffffff  
mov    esi, eax  
xor    esi,0xffffffff  
or     esi,ecx  
or     eax,ecx  
sub    eax,esi  
add    eax,edx  
mov    ecx,DWORD PTR [rbp-0x4]  
mov    DWORD PTR [rbp-0x10],eax  
mov    eax,ecx  
cdq  
mov    ecx,DWORD PTR [rbp-0x10]  
idiv  ecx
```

$$x \wedge y \dashrightarrow (x \mid y) - (\sim x \mid y) + (\sim x)$$



MOVfuscator

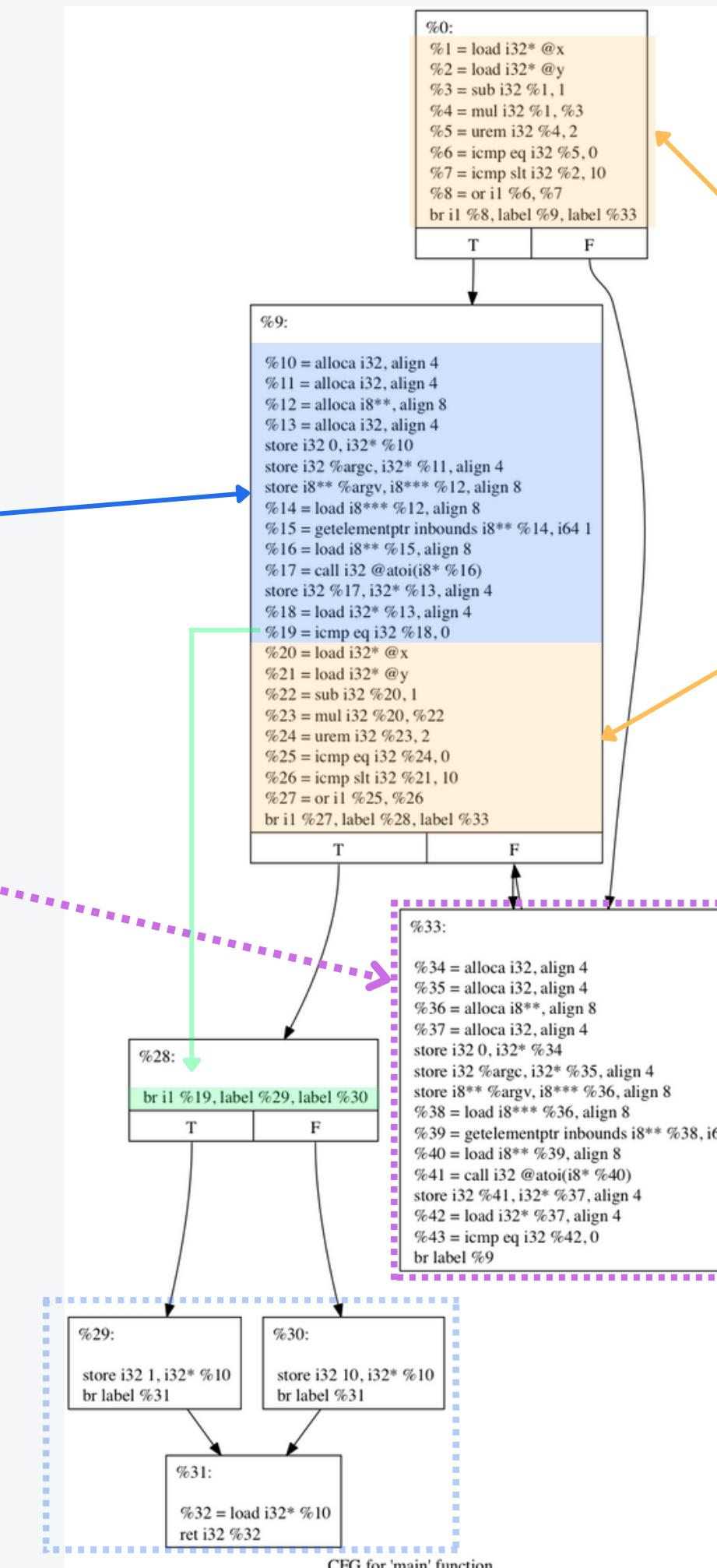
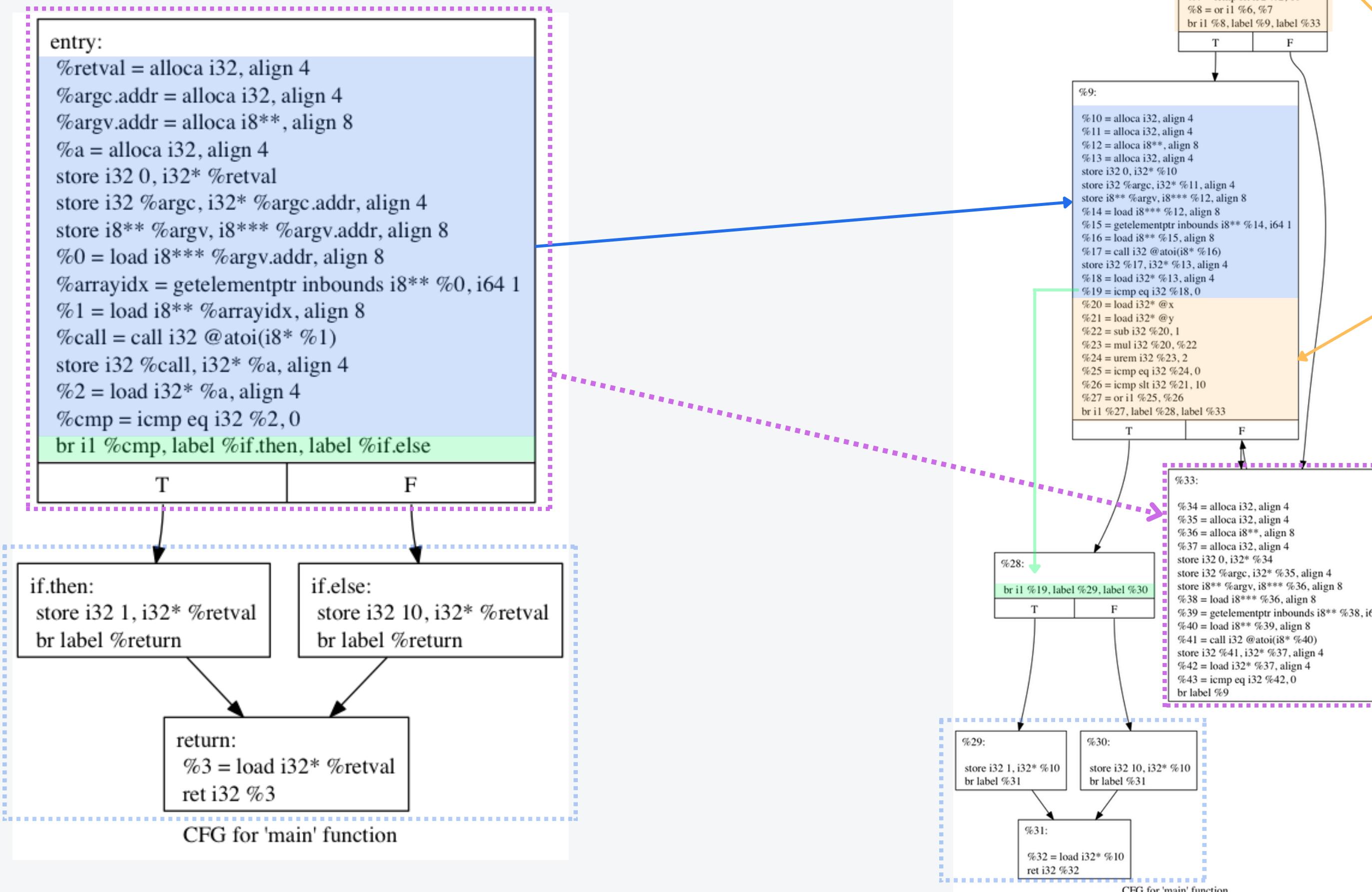
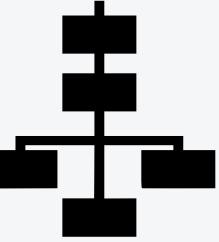
```
<is_prime>:  
    push    ebp  
    mov     ebp,esp  
    sub    esp,0x10  
    cmp    DWORD PTR [ebp+0x8],0x1  
    jne    8048490 <is_prime+0x13>  
    mov    eax,0x0  
    jmp    80484cf <is_prime+0x52>  
    cmp    DWORD PTR [ebp+0x8],0x2  
    jne    804849d <is_prime+0x20>  
    mov    eax,0x1  
    jmp    80484cf <is_prime+0x52>  
    mov    DWORD PTR [ebp-0x4],0x2  
    jmp    80484be <is_prime+0x41>  
    mov    eax,DWORD PTR [ebp+0x8]  
    cdq  
    idiv   DWORD PTR [ebp-0x4]  
    mov    eax,edx  
    test   eax,eax  
    jne    80484ba <is_prime+0x3d>  
    mov    eax,0x0  
    jmp    80484cf <is_prime+0x52>  
    add    DWORD PTR [ebp-0x4],0x1  
    mov    eax,DWORD PTR [ebp-0x4]  
    imul   eax,DWORD PTR [ebp-0x4]  
    cmp    eax,DWORD PTR [ebp+0x8]  
    jle    80484a6 <is_prime+0x29>  
    mov    eax,0x1  
    leave  
    ret
```



Vkladanie kódu

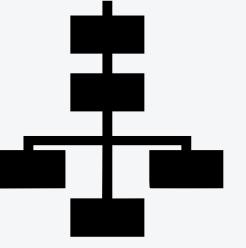
- Nadbytočný kód (garbage code)
- Nedosiahnutelný kód (dead code)
- "Jednoduchý" spôsob ako oklamat' AV
- Optimalizácia?

Bogus Control Flow



predikáty

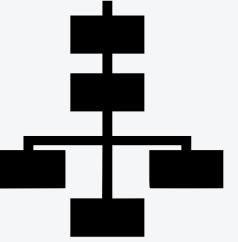
klon



Obfuskačné predikáty

(Opaque predicates)

- Základ mnohých transformácií
 - BCF, CFG Flatenning
- Mnoho spôsobov konštrukcie
- Zvyčajne invariantné



Obfuskačné predikáty #1

foo(x)

rand()

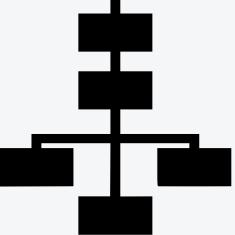
$$0 \neq (x^2 + x + 7) \% 81$$

True

False



Obfuskačné predikáty #2



foo(x)

r = x % arr_size

idx = arr1[r]

var = arr2[idx]

arr1 = [0, 1, 2, 3, 4, 5, 6]

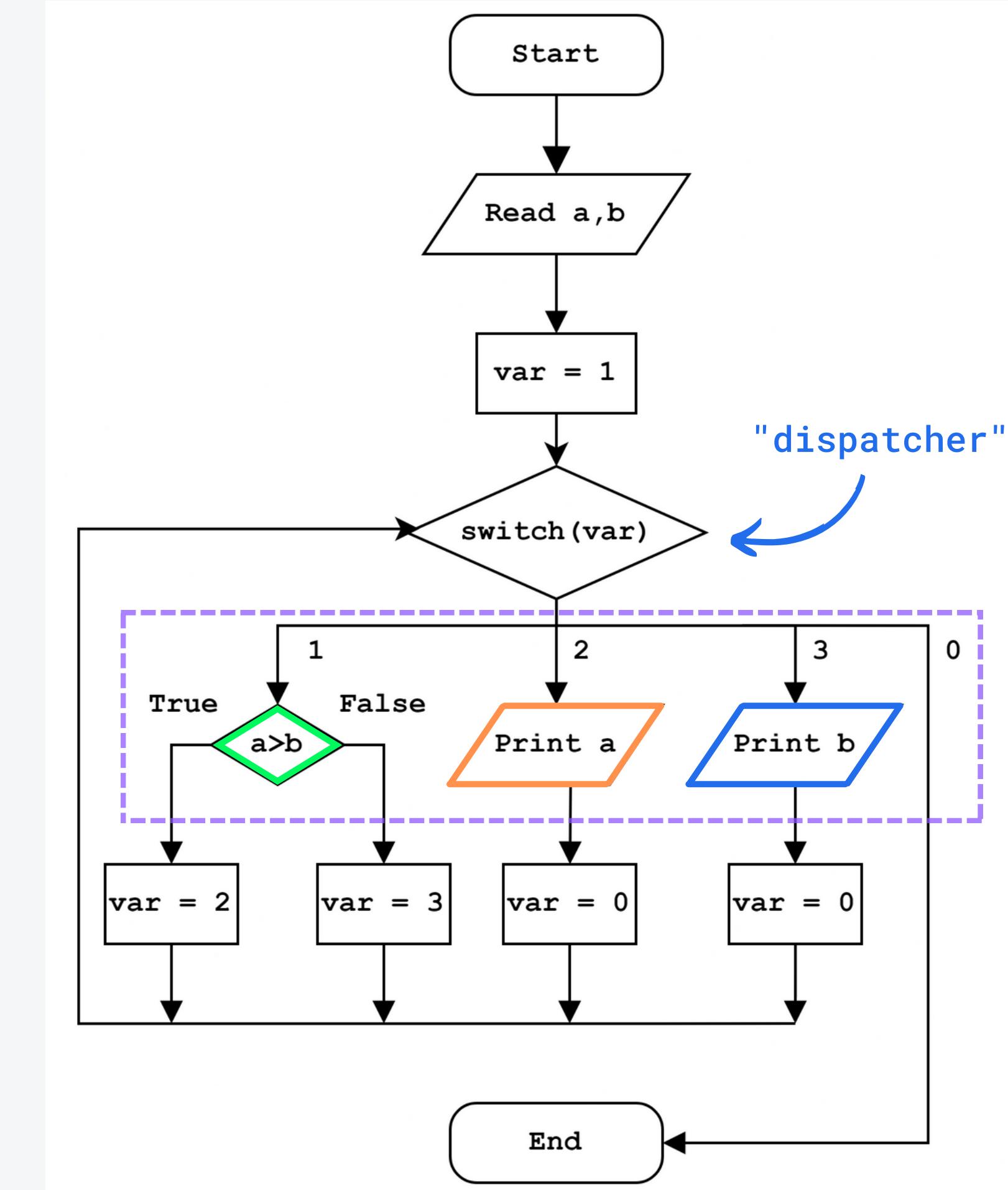
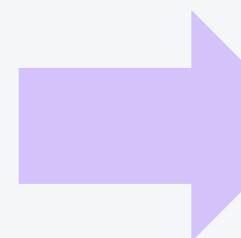
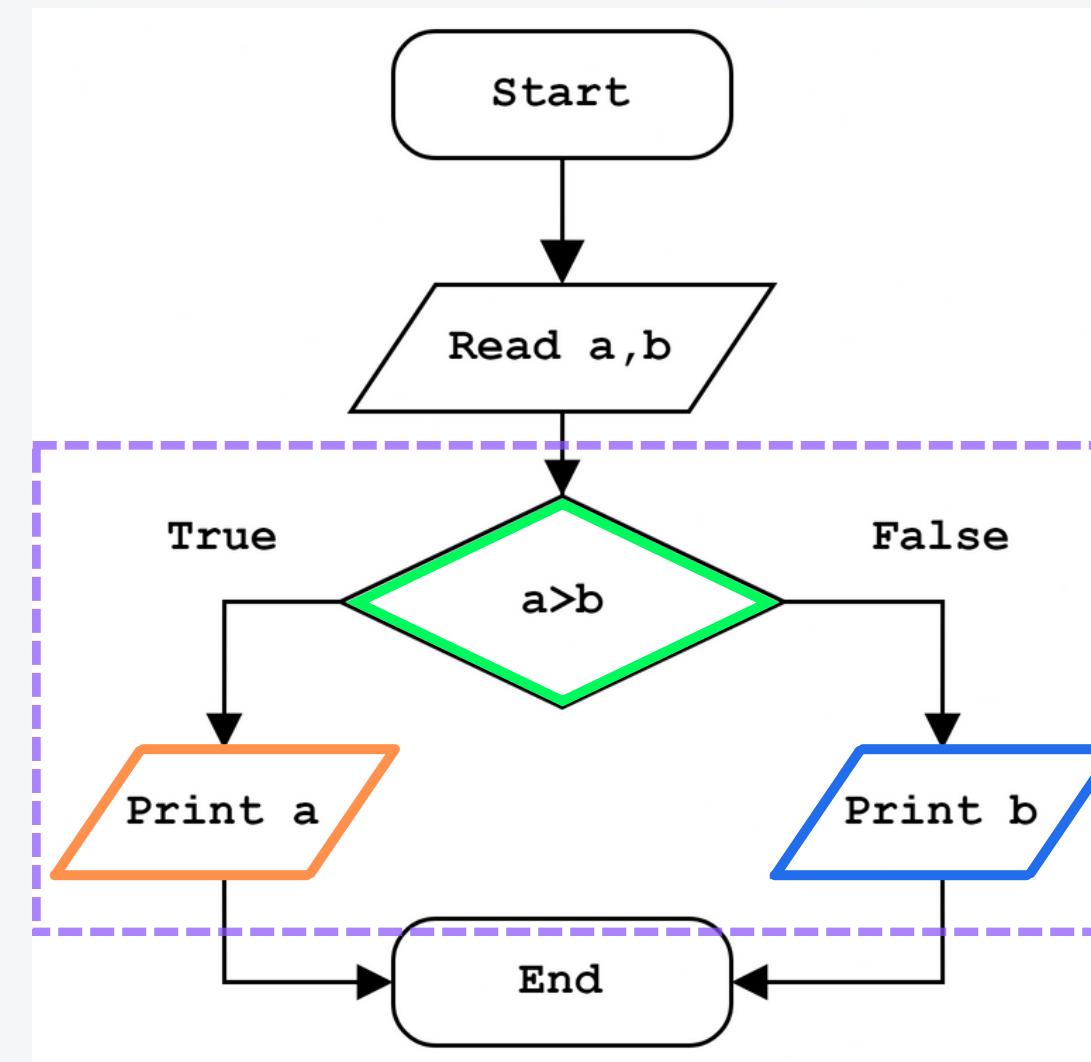
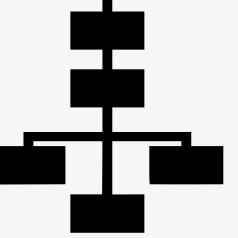
arr2 = [6, 5, 4, 3, 2, 1, 0]

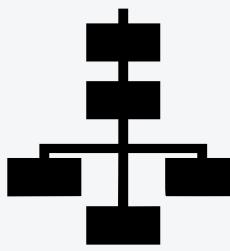
var == ...

True

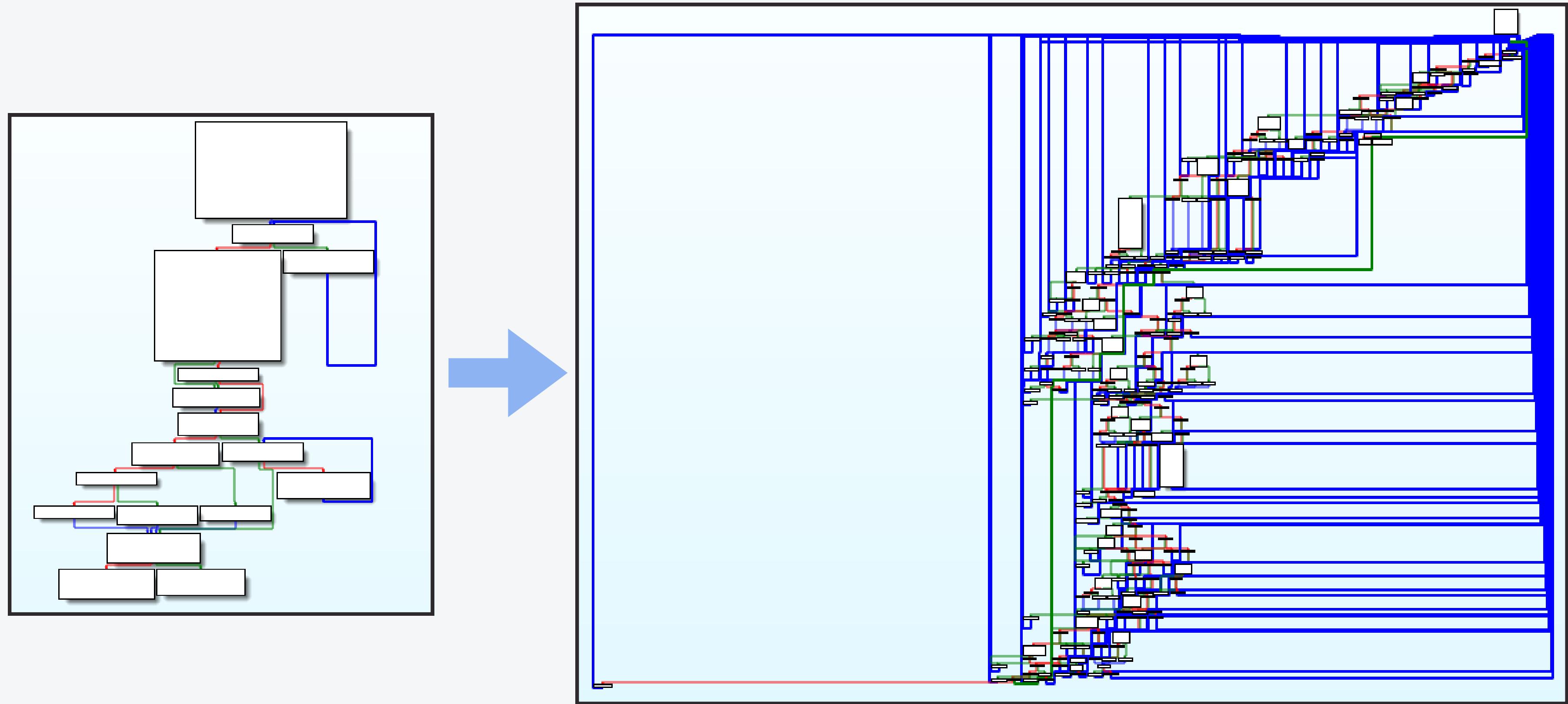
False

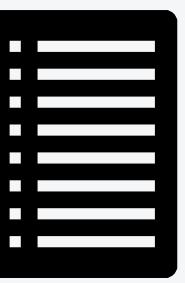
Control Flow Flattening





BCF + Flattening



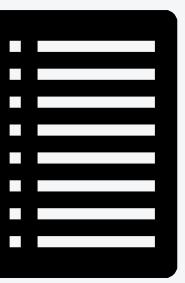


Obfuskácia konštánt

$$\left. \begin{array}{l} f(x) = ax + b \\ g = f^{-1} \end{array} \right\} g(f(x)) = x$$

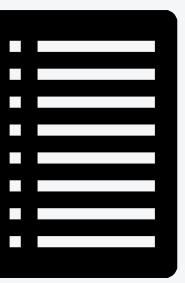
$$E = x + y - (x \mid\mid y) - (!x \mid\mid y) + (!x) = 0$$

$$\begin{aligned} C &= g(x + y - (x \mid\mid y) - (!x \mid\mid y) + (!x) + f(C)) \\ C &= g(0 + f(C)) \\ C &= g(f(C)) = C \end{aligned}$$



Obfuskácia reťazcov

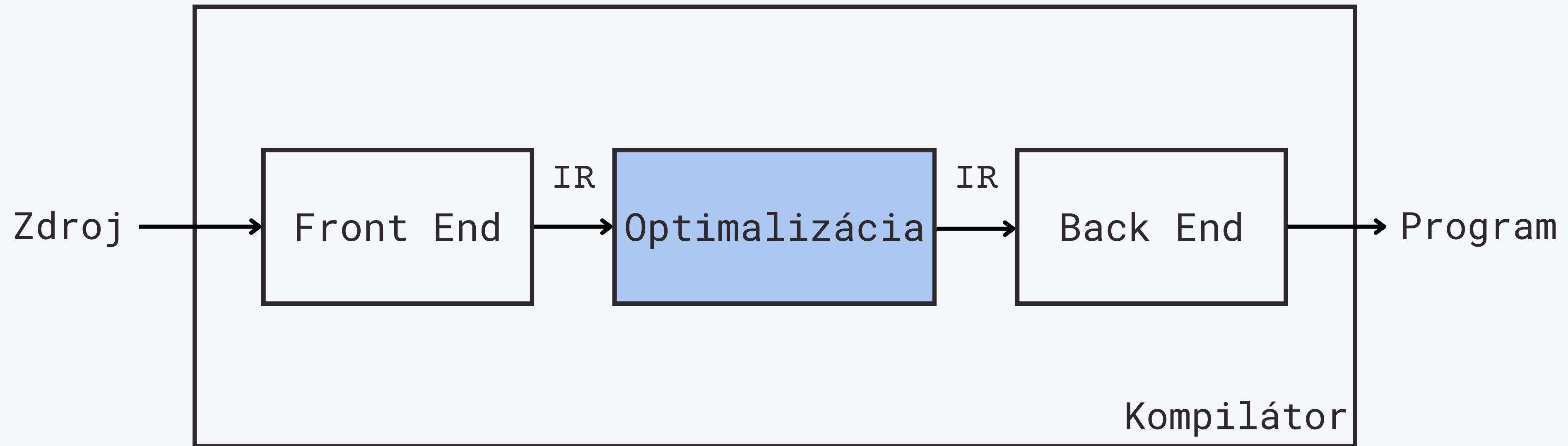
- Textové reťazce sa zakódujú počas komplilácie
- Do programu sa vloží dekódovacia funkcia
- Najčastejšie XOR
- Vhodné kombinovať s ďalšími transformáciami



Diverzifikácia

- Použitie PRNG
- Seed --> determinizmus
 - Vhodné pre testovanie
- Substitúcie, náhodne vybrané bloky, predikáty
- Náhodný kľúč pre kódovanie reťazcov

LLVM a obfuskácia



LLVM IR

```
define i32 @foo() {
entry:
    %add = add i32 7, 42
    %cmp = icmp sgt i32 %add, 56
    br i1 %cmp, label %if.then, label %if.end

if.then:
    br label %return

if.end:
    br label %return

return:
    %retval.0 = phi i32 [7, %if.then], [%add, %if.end]
    ret i32 %retval.0
}
```

LLVM a obfuskácia

- In-tree vs. out-of-tree LLVM Pass
- Generovanie IR a aplikácia transformácií:

```
$ clang -S -emit-llvm source.c --> *.ll  
$ opt -S -load libMyPass.so -my-pass-flag *.ll -o out.ll  
$ clang out.ll -o my_binary
```

- Generovanie CFG pomocou "**opt -dot-cfg**"
- Ďalšie užitočné prepínače:
 - disable-O0-optnone --> použitie **opt** aj pri **-O0**
 - fno-discard-value-names --> zachová názvy premenných

Open-source LLVM obfuskátory

- **0-LLVM**
 - BCF, Flattening, Substitúcia
- **Armariris**
 - Reťazce
- **Hikari**
 - AntiClassDump, Objective-C, volania funkcií
- **Kryptonite**
 - PoC, anti-debug

0-LLVM - <https://github.com/obfuscator-llvm/obfuscator>

Hikari - <https://github.com/HikariObfuscator/Hikari>

Armariris - <https://github.com/GoSSIP-SJTU/Armariris>

Kryptonite - <http://Overcl0k.tuxfamily.org/bl0g/?p=260>

Vlastná implementácia

- Odolné predikáty v BCF
- MBA substitúcie
- Kódovanie reťazcov ľubovoľnou funkciou
- Obfuscácia konštánt

Metriky

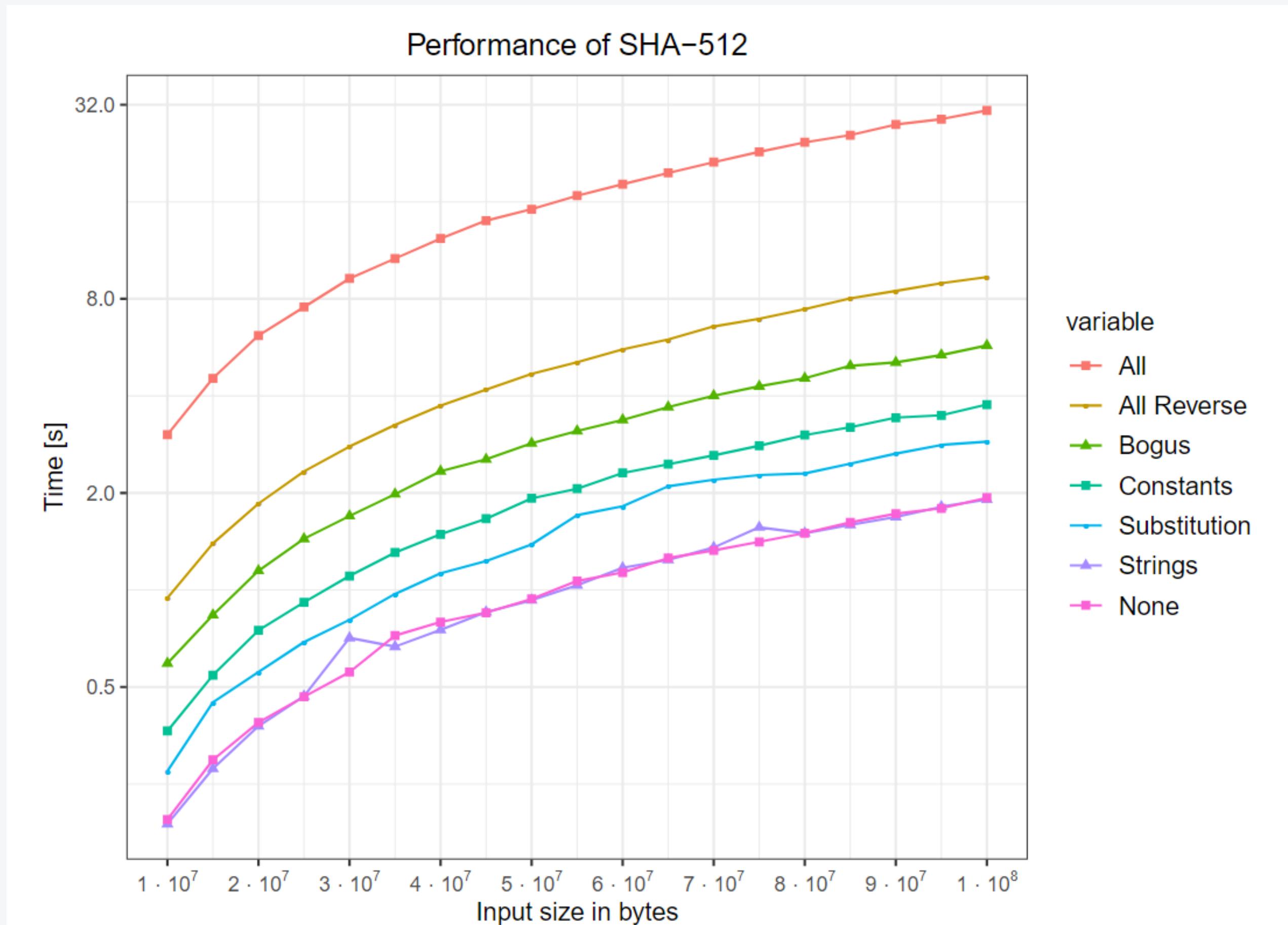
- **Odolnosť** (Resilience)
 - číselné metriky
- **Potencia**
 - zhľadanie automatického deobfuscátora
- **Výkon** (Cost)
 - benchmark

Potencia

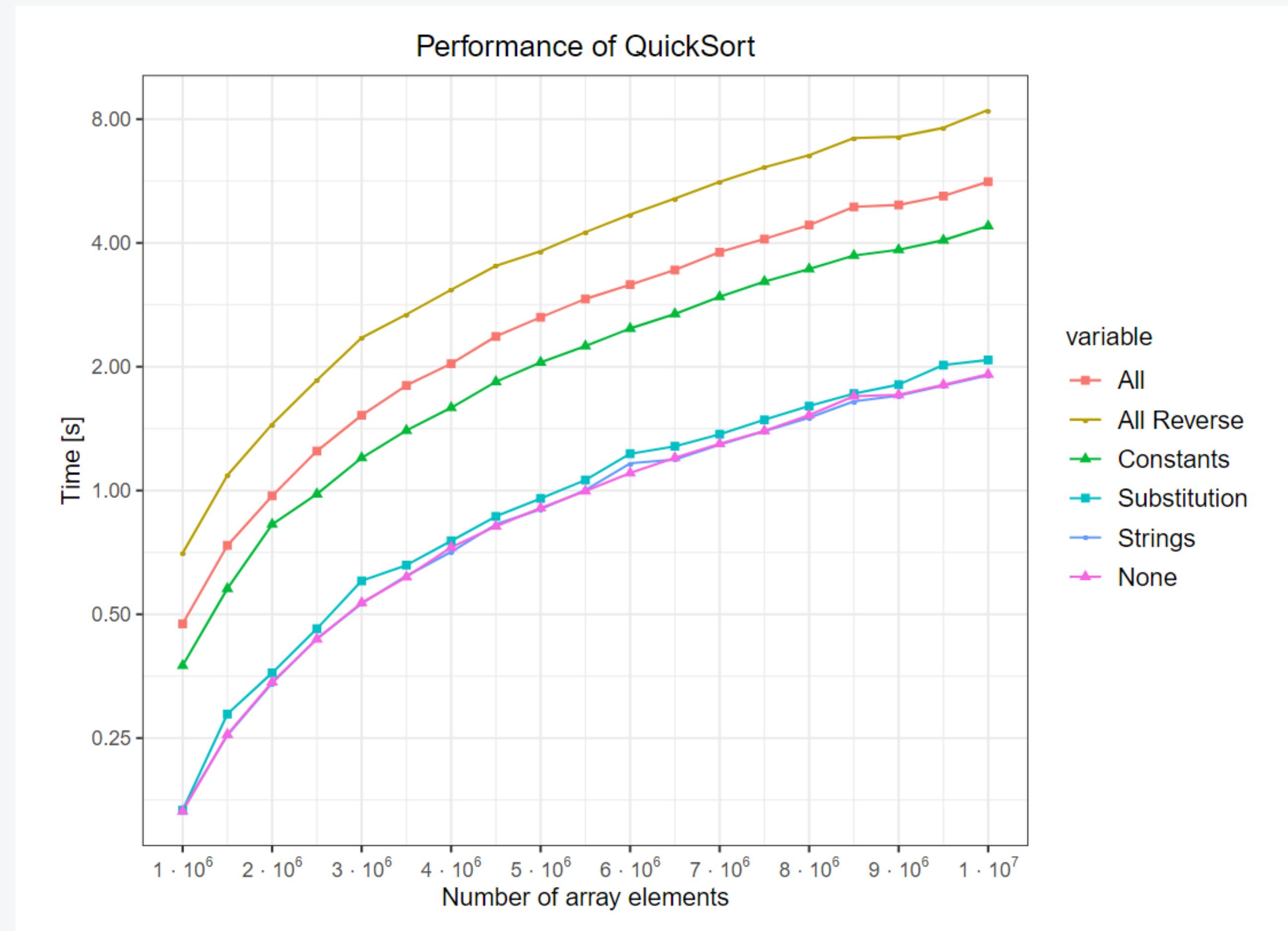
μ_1 - Počet inštrukcií
 μ_2 - Kolmogorova zložitosť

Program	SHA-512		AES		QuickSort	
Metric	μ_1	μ_2	μ_1	μ_2	μ_1	μ_2
Substitution	1.37	1.07	1.09	1.04	1.23	1.03
Substitution $\times 2$	2.25	1.26	1.34	1.13	1.82	1.11
Opaque constants	2.66	1.86	3.29	2.40	3.76	1.96
String obfuscation 1	1.05	1.04	1.01	1.02	1.19	1.12
String obfuscation 2	1.09	1.05	1.02	1.03	1.35	1.16
Bogus Control Flow	3.52	2.23	4.28	3.49	7.18	3.40
Str 2 + Bogus	3.72	2.35	4.40	3.64	10.64	4.44
Str 2 + Bogus + Const	14.26	8.08	19.16	16.23	32.43	12.27
All	26.98	11.03	36.05	23.12	57.88	16.23
Reverse order	8.10	4.24	9.50	7.05	21.57	7.84

Vplyv na výkon



Vplyv na výkon



Výhody a nevýhody využitia LLVM

- Podporované jazyky
 - C, C++, Go, Haskell, Rust, Swift...
- Podporované architektúry
 - **x86**, **ARM**, PowerPC, MIPS...
- Rozsiahle API
- Kompatibilita verzií



Otázky?

romanoravec1@gmail.com