

Санкт-Петербургский государственный политехнический университет  
Кафедра прикладной математики

# Построение набора регулярных грамматик для реализации генератора виртуальных машин на произвольном вычислительном устройстве

Диссертация на соискание ученой степени магистра

Выполнила студентка гр.6057/2  
Научный руководитель

Кононова А.Г.  
Аранов В.Ю

# Определения

- Реинжиниринг (англ. *reengineering*) – обратное проектирование
- Обфускация (англ. *obfuscation*) – совокупность методов и средств, направленных на затруднение анализа и реинжиниринга программного кода
- Дизассемблирование – перевод программы из машинного кода на язык ассемблера

# 1. Цели и задачи

## Цель работы

- Разработка методов защиты программ от реинжиниринга

## Решаемые задачи

- Обфускация кода с помощью контекстно-независимых грамматик
- Обфускация кода на базе сети Петри

## 2. Построение формальной грамматики

- Множество x86 инструкций – рабочий алфавит
- Правила вывода определяются формальной грамматикой
- Ассемблер – язык, генерируемый такой грамматикой
- Терминальные символы включают в себя «мусорные команды»

### 3. Обфускация и формальные грамматики

*Исходный код*

```
mov eax, 3  
mov a, eax
```

*Правила вывода*

```
S ::= aA | bA | xB | xC  
A ::= cA | xC  
B ::= dB | aC | y  
C ::= aB | bB
```



*Терминальные символы*

```
x = mov eax, α  
y = mov β, eax
```

```
a = add reg1, γ; sub reg1, γ  
b = push reg2; pop reg2  
c = inc ecx  
d = xchg ecx, ebx
```

*Обфусцированный код*

```
add ebx, 5  
sub ebx, 5  
inc ecx  
mov eax, 3  
add edx, γ  
sub edx, γ  
xchg ecx, ebx  
add edx, 9  
sub sub edx, 9  
push acx  
pop acx  
add edx, γ  
sub edx, γ  
xchg ecx, ebx  
xchg ecx, ebx  
mov a, eax
```

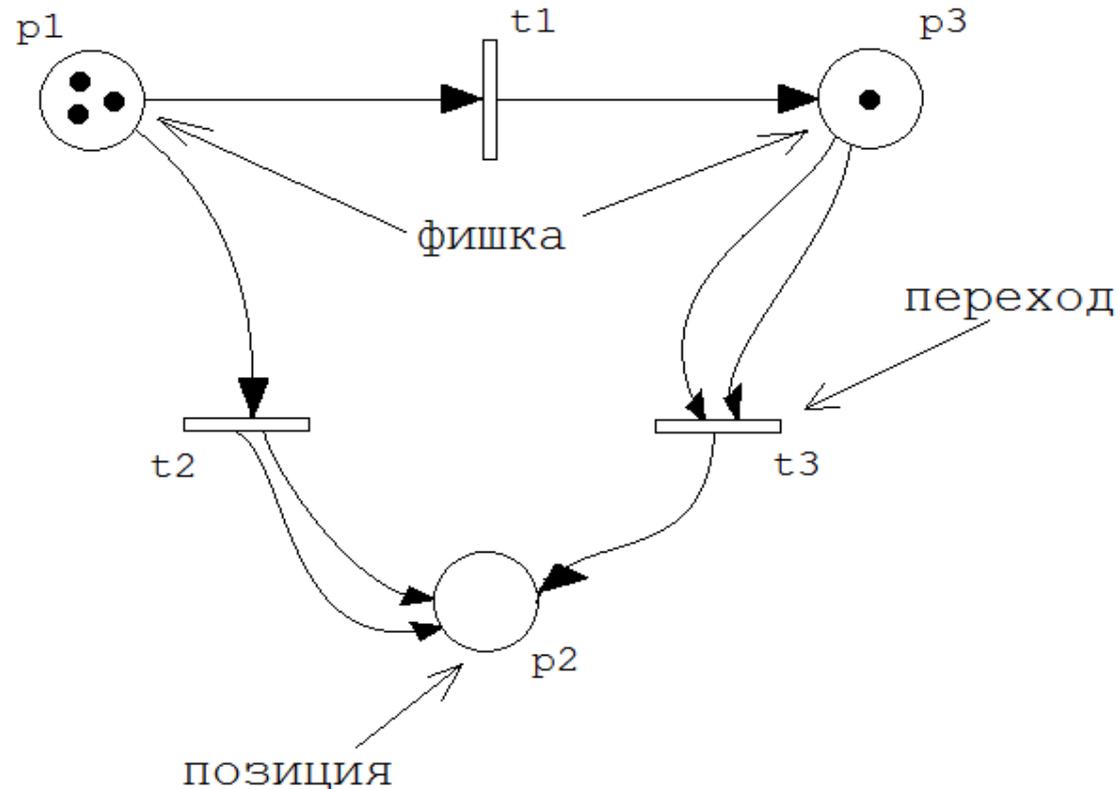
## 4. Набор функциональностей для защиты, используемый в коммерческих приложениях

Протектор	VM	RVM	Anti-Dump	ZPerm	GC
FlexNet	нет	нет	да	нет	нет
WinLicence	да	да	да	да	да
FrontLine	да	да	да	да	да
VMProtect	да	да	да	нет	да
Armadillo	да	нет	да	нет	нет
ExeCrypter	да	нет	да	нет	нет
ASProtect	нет	нет	да	нет	нет
Obsidium	нет	нет	да	да	да

# 5. Сеть Петри

$$C = (P, T, I, O)$$

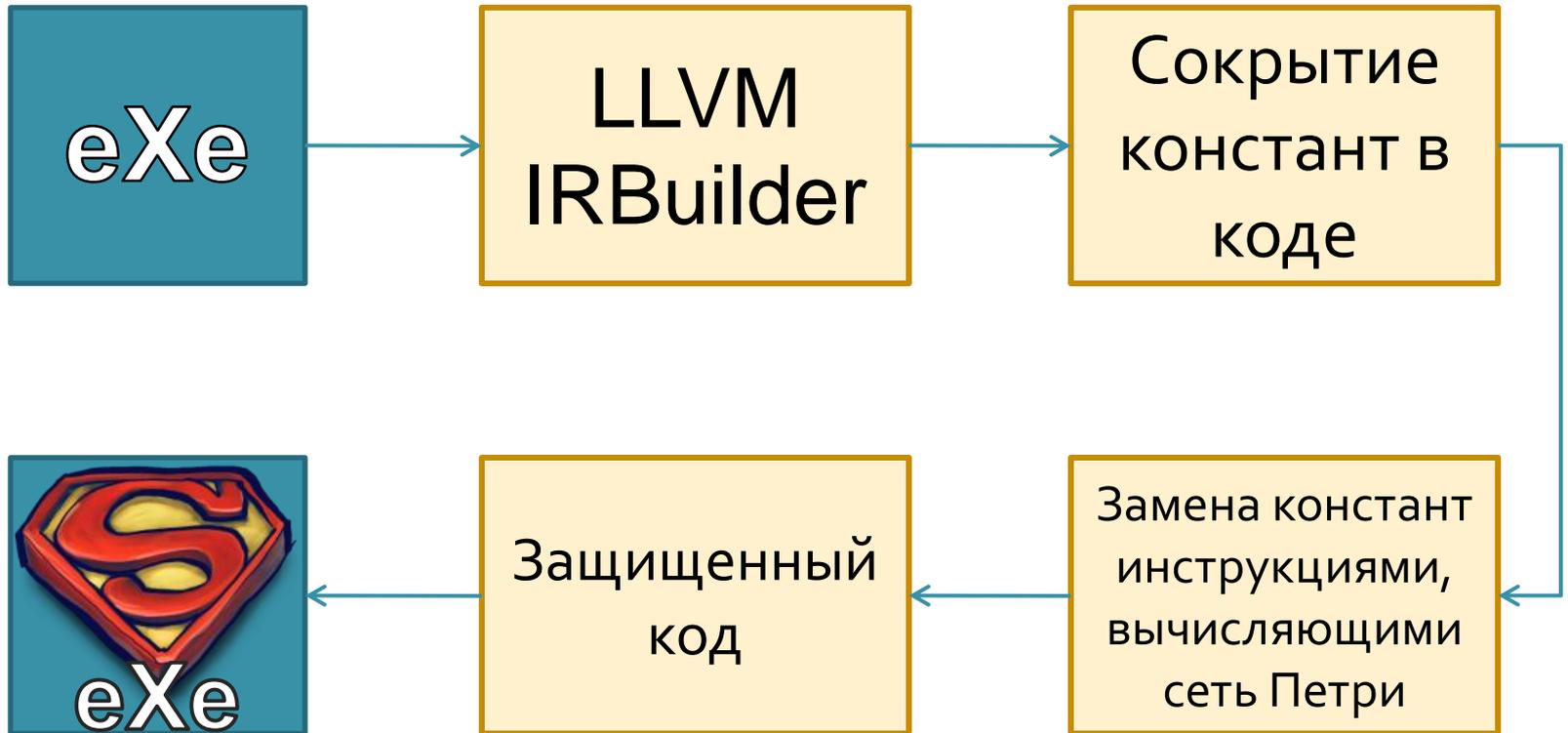
- $P$  - множество позиций,  $T$  - множество переходов,  $I$  – входная функция,  $O$  - выходная функции
- $\mu$  - маркировка,  $\mu: P \rightarrow N$ ,  $R(C, \mu)$  – множество достижимости  $\mu$



## 6. Задача достижимости и построение сети Петри

- Выполняется ли для данной  $\mu' : \mu' \in R(C, \mu)$ ?
- «На сегодняшний день не существует ни алгоритма, решающего задачу достижимости, ни доказательства того, что такого алгоритма не может быть». Питерсон Дж. Теория сетей Петри и моделирование систем. М, Мир, 1984. стр. 145
- Задача построения сети Петри разрешима за полиномиальное время

# 7. Обфускация на базе сети Петри



## 8. Результаты: варьирование параметров сети

### *Параметры сети*

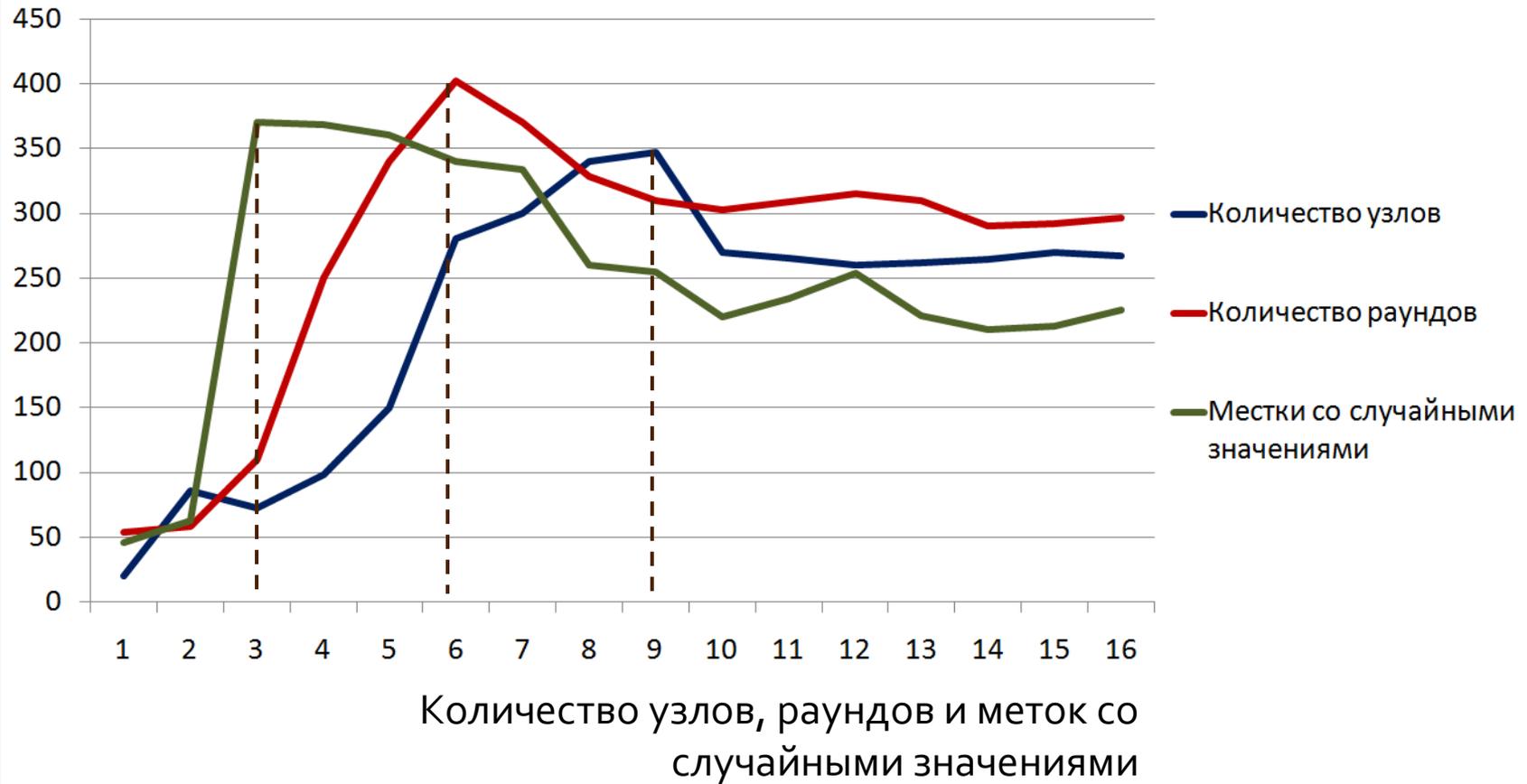
- количество узлов
- количество раундов
- количество меток со случайными значениями

### *Характеристики исходного файла*

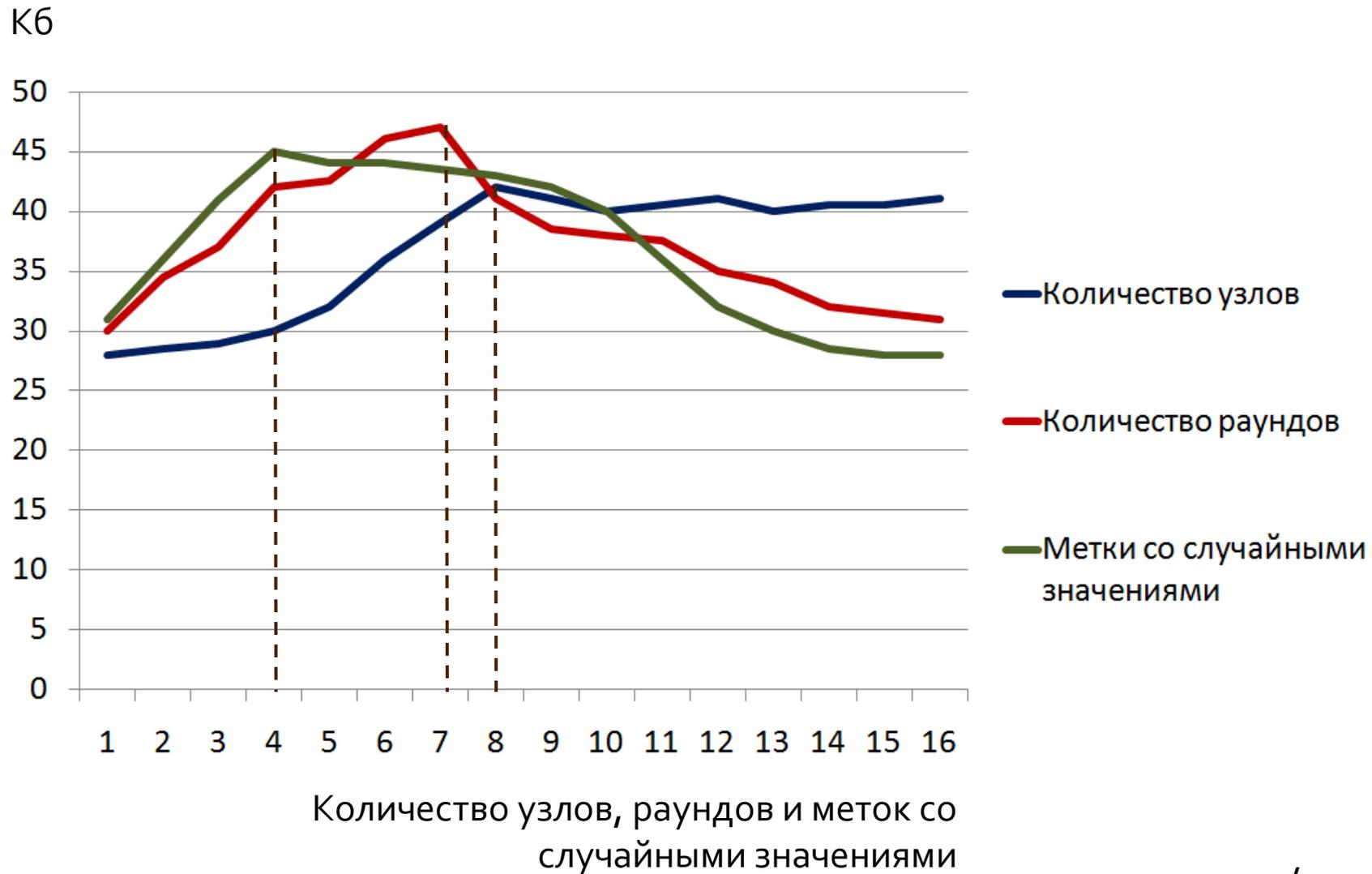
- размер файла – 27Кб
- время выполнения – 4,37 мкс
- количество строк кода – 5
- количество констант в коде - 4

# 8.1. Количество строк кода

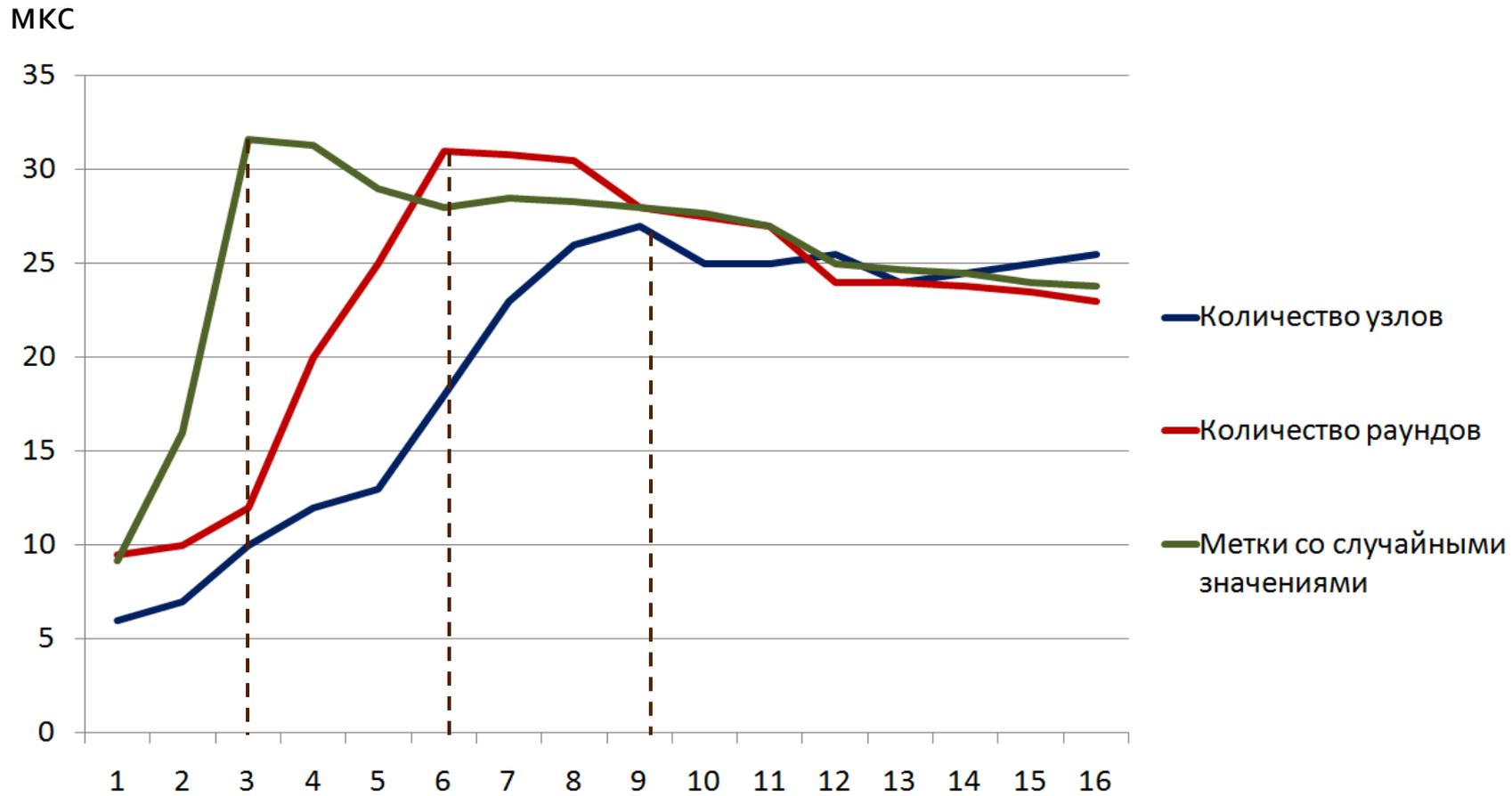
Строки  
кода



## 8.2. Размер исполняемого файла

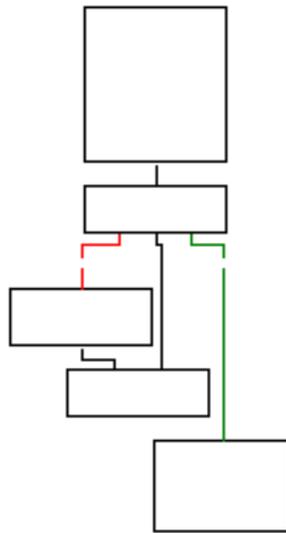


## 8.3. Время выполнения файла

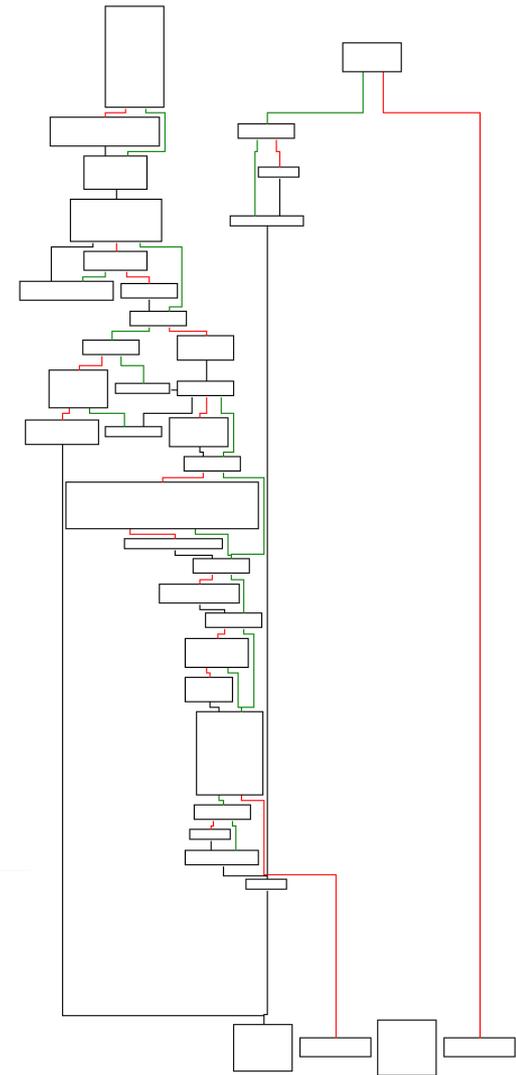


Количество узлов, раундов и меток со случайными значениями

# 9. Результаты: блок-схемы подпрограмм



Блок-схема оригинальной подпрограммы



Блок-схема обфусцированной подпрограммы

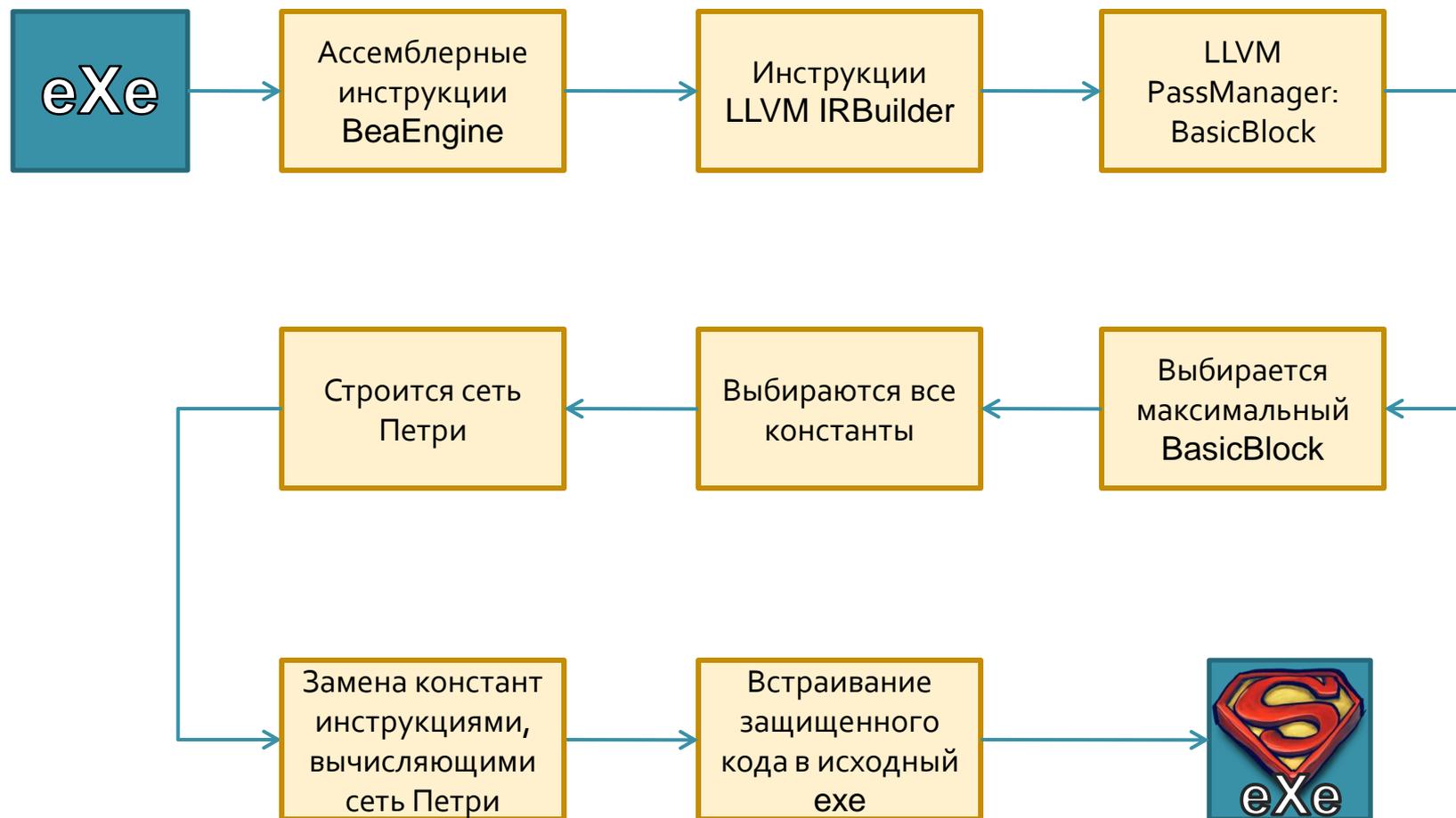
# 10. Заключение

- Предложена и реализована обфускация с помощью контекстно-независимых грамматик
- Одна грамматика порождает около 1000 различных вариантов ассемблерного кода, общее число вариаций более  $4 * 10^8$
- Предложена и реализована обфускация на базе сети Петри
- Защита на базе сетей Петри, не используется в известных коммерческих продуктах (ноябрь 2011 года)



Спасибо за внимание

# Обфускация на базе сети Петри: подробная схема



# Построение сети Петри: узлы

**Const**

**Rand**

**?**

# Построение сети Петри

- каждый узел имеет двух родителей
- для неизвестных узлов строится СЛДУ
- вычисляются неизвестные узлы и строится сеть
- константа заменяется инструкцией, вычисляющей сеть
- защищенный код не содержит константы