



ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
та
КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

3(10)2007

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

МІЖНАРОДНИЙ НАУКОВО-ТЕХНІЧНИЙ ЖУРНАЛ

Засновано у 2004 році
Виходить тричі на рік
№ 3(10), 2007

Зареєстрований Державним комітетом інформаційної політики,
телебачення та радіомовлення України.
Свідоцтво про реєстрацію № 9007, сер. КВ від 27.07.2004 р.

Рекомендовано до друку рішенням Ученої ради ВНТУ, протокол № 6 від 29.11.2007 р.

Міжнародний науково-технічний журнал „Інформаційні технології та комп'ютерна інженерія” є науковим фаховим виданням України, в якому можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора та кандидата технічних наук (постанова ВАК України №2-05/1 від 19.01.2006 року)

© Вінницький національний технічний університет

Адреса редакції:
Україна, 21021, м.Вінниця,
вул. Хмельницьке шосе, 95
ВНТУ, к. 2220

Тел: +380 (432) 43-90-02
Факс: +380 (432) 46-57-72
E-mail: itce@vstu.vinnica.ua
<http://vstu.vinnica.ua/itce/>

ТЕОРІЯ ПРИЙНЯТТЯ РІШЕНЬ

ОПТИМІЗАЦІЯ МОДЕЛІ ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ 3 127
ВИКОРИСТАННЯМ GARCH-МЕТОДІВ

Р.Н. Квстний, В.В. Кабачій, В.Ю. Коцюбинський, Л.М. Кислиця
Вінницький національний технічний університет

ИНФОРМАЦИОННОЕ ПРОСТРАНСТВО И КЛАССИФИКАЦИЯ 131
СТРАТЕГИЙ УПРАВЛЕНЧЕСКОЙ ДЕЯТЕЛЬНОСТИ В ТЕОРИИ
ИГР И ПРИНЯТИЯ РЕШЕНИЙ

А.А. Шиян
Вінницький національний технічний університет

ГАРАНТОЗДАТНІ (НАДІЙНІ І БЕЗПЕЧНІ) СИСТЕМИ ТА ТЕХНОЛОГІЇ

ОПТИМИЗАЦИЯ ПРОЦЕССА ПОИСКА ПРИЧИНЫ ОТКАЗА РЭА В 140
УСЛОВИЯХ ДЕФИЦИТА ИНФОРМАЦИИ

А.Л. Овчинников, Г.Г. Сергеев
Севастопольский Национальный технический университет

ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ МОНІТОРИНГУ ТА 144
КОНТРОЛЮ ОПТИЧНИХ ВОЛОКОН

С.В. Павлов, Р.В. Просоловський
Вінницький національний технічний університет

ЗАХИСТ ІНФОРМАЦІЇ

МОДУЛЬНЕ ПРОЕКТУВАННЯ ОПТИМАЛЬНОЇ СИСТЕМИ 155
ЗАХИСТУ ІНФОРМАЦІЙНИХ РЕСУРСІВ

А.В. Дудатьєв
Вінницький національний технічний університет

ЗАХИСТ MAPLE-ПРОЦЕДУР 159

В. М. Михалевич, І.В. Димніч, О. В. Михалевич
Вінницький національний технічний університет

МЕТОД ОЦЕНКИ СТЕПЕНИ ОРГАНИЗОВАННОСТИ СИСТЕМ 166
ЗАЩИТЫ ИНФОРМАЦИИ

Н.И. Алишов, И.А. Жуков, В.А. Марченко
Институт кибернетики НАН Украины
Национальный авиационный университет

УДК 681.3.06

ЗАХИСТ MAPLE-ПРОЦЕДУР

В. М. Михалевич, І. В. Димніч, О. В. Михалевич

Вступ

В попередніх статтях автора [1-7] була поставлена задача створення в середовищі системи комп'ютерної алгебри Maple навчально-контролюючого комплексу з дисциплін математичного спрямування. Очевидно, що процедури подібного комплексу повинні мати захист. По – перше для блокування доступу до навчальних процедур в режимі тестування, а по – друге для унеможливлення несанкціонованого перегляду та копіювання програмного коду розроблених процедур. Відсутність механізму захисту Maple-процедур може привести до створення програм-клонів і тим самим руйнації функції контролю розроблюємого навчально-контролюючого комплексу. Проте, у жодному з літературних джерел [0-0], у тому числі і в документації [0, 0] та довідковій системі Maple, автори не знайшли описання можливих прийомів подібного захисту.

Мета статті

Метою даної роботи є розробка технології захисту Maple-процедур від несанкціонованого доступу та перегляду.

Постановка проблеми

Відсутність задокументованих прийомів захисту Maple-процедур пояснюється тим, що в самій організації цієї системи в значній мірі реалізовано принцип відкритості. Частина процедур написана на мові С і є ядром, яке складає лише до 10% загального розміру системи. Решта програм, що реалізують функціональність системи, написані на власній мові програмування системи Maple. Всі ці програми розташовані в основній бібліотеці або в одному з чисельних спеціалізованих пакетів. Програмний код всіх процедур, що написані на мові Maple доступні для перегляду та модифікації! До того ж, розробники системи Maple радять досвідченим користувачам ознайомитись з такими процедурами для кращого опанування програмування в середовищі цієї системи [13].

Для визначення належності певної процедури до ядра системи потрібно записати наступний вираз:

```
> print(`diff`);
```

```
proc() option builtin, remember; 185 end proc
```

Наявність опції `builtin` означає, що програма `diff`, яка призначена для знаходження похідної, є вбудованою, тобто одною із процедур, що написані на мові С, скопійовані та розміщені в ядрі системи. Переглянути програмний код таких процедур стандартними засобами Maple неможливо. Як і неможливо створити власну вбудовану процедуру.

Перевіримо команду інтегрування:

```
> print(`int`);
```

```
proc() ... end proc
```

Відсутність опції `builtin` означає, що дана процедура написана на мові Maple. Система розпізнала дану команду як процедуру, тобто команда `int` розміщена в основній бібліотеці.

Щоб проглянути програмний код будь-якої Maple-процедури, що створена користувачем, потрібно виконати команду:

```
> print(`proc_name`);
```

де `proc_name` – ім'я потрібної Maple-процедури. Результатом виконання команди буде виведення на екран монітора тексту указаної процедури:

```
> f:= proc(x) x^2 end proc;
```

```
> print(f);
```

```
proc(x) x^2 end proc
```

Для захисту відображення в області виведення результатів обчислень операторів тіла процедури користувача призначена опція `Copyright`:

```
> f:= proc(x) option Copyright; x^2 end proc;
> print(f);

proc(x) ... end proc
```

Але такий захист просто знімається зміною значення змінної інтерфейсу `verboseproc` за допомогою команди `interface` (за умовчуванням: `verboseproc=1`)

```
> interface(verboseproc=2) :
> print(f);

f:= proc(x) option Copyright; x^2 end proc
```

Установка відповідного значення змінної `verboseproc` рівним або більшим за 2 дозволяє переглянути програмний код будь-якої процедури системи, що написана на мові Maple i, отже, не є вбудованою:

```
> print(`int`);
proc ()
local answer;
option `Copyright (c) 1997 Waterloo Maple Inc. All rights reserved.`;
if nargs = 2 and type(args[2],
`=('name', '{ complex(float) .. algebraic, algebraic .. complex(float) }')) then
try
answer:= evalf('int'(args));
if type(answer, 'complex(float)') then return answer end if
catch :
end try
end if;
answer := `int/int`([args], Digits, _EnvCauchyPrincipalValue, _EnvAllSolutions,
_EnvContinuous);
if answer = FAIL then error "wrong number (or type) of arguments" end if;
answer
end proc
```

Процедури, що створено користувачем можна записувати за допомогою команди `save` у файли з розширенням `m` [8, 11, 12], що мають спеціальний внутрішній Maple-формат. Безпосередньо, наприклад, за допомогою текстового редактора, неможливо переглянути програмний код Maple-процедур, що записані у такий файл. Для користування записаними процедурами в данному сеансі роботи з системою Maple потрібно за допомогою команди `read` зчитати відповідний файл. Після цього збережені Maple-процедури стають не тільки доступними для використання, а й для перегляду їх програмного коду описаними вище прийомами. Отже, в разі завантаження певної Maple-процедури в пам'ять комп'ютера користувач має доступ до тексту цієї процедури, незалежно від того, де знаходиться файл з цією процедурою.

Розв'язання проблеми

Проаналізуємо нові можливості системи Maple з точки зору розробки механізму захисту процедур, розроблюємих користувачем.

Починаючи з версії Maple 6, яка з'явилася в грудні 2000 р. [0] систему наділено рядом нових можливостей в програмуванні, які періодично вдосконалюються та доповнюються [0], зокрема введено:

Нову конструкцію `module` для реалізації абстрактного типу даних на основі інкапсуляції – об'єднання даних та процедур їх обробки [0, 0, 0].

Засоби для виклику з середовища Maple зовнішніх процедур, що написані на мові C, Fortran або Java (External Calling).

Засоби доступу до Maple-процедур в програмах, що створюються на мові C або C++ (OpenMaple).

Кожна з указаних можливостей заслуговує на увагу з точки зору розробки механізму захисту Maple-процедур. В даній роботі досліджуються можливості конструкції **module**.

Модулі надають мові програмування Maple деякі властивості мов об'єктно-орієнтованого програмування [0, 0, 0]. Розглянемо створення модуля на прикладі задачі відтворення процесу знаходження похідної. Як за синтаксисом визначення так і за своєю структурою модуль здебільшого схожий на процедуру. Оголошення модуля починається зі слова **module** та завершується ключовими словами **end module**:

```
> _Dif:=module()
  description "Символьне диференціювання степеневих функцій + правила
  диференціювання суми та добутку";
  local show_dif, d_pow, `Зчитування_пароля`, `пароль_користувача`,
  `пароль_клавіатура`;
  export s_diff;
  option package;
  Тіло модуля
end module;
```

В модулі так само як і в процедурі може використовуватися оголошення локальних (**local**) та глобальних (**global**) змінних, рядок описання (**description**), задання опцій (**option**).

Разом з тим модулі мають ряд суттєвих відмінностей, зокрема:

- модуль не має списку вхідних параметрів;
- набір опцій в модулі дещо інший, зокрема введена можливість створення пакетів, що представляють собою набір процедур і доступ до яких забезпечується командою **with**;
- модуль має оператор **export**, що оголошує змінні, які експортуються в зовнішнє середовище.

Виключне значення для конструкції модуль має оператор **export**. В цьому операторі описуються імена змінних та процедур, які модуль експортує в зовнішнє середовище. Ці імена є певним компромісом між локальними та глобальними: з одного боку вони як і локальні не доступні для безпосередньої зміни користувачем, а з іншого – це механізм за допомогою якого користувач має не тільки доступ до оперування із значеннями локальних змінних, а й до зміни їх значень за допомогою процедур модуля, що експортуються.

Тіло модуля **_Dif** утворюють процедури **show_dif**, **d_pow**, **`пароль_файл`**, **`пароль_клавіатура`** та **s_diff**.

Локальна процедура **d_pow** призначена для знаходження похідної виразу, що задається користувачем; локальна процедура **show_dif** – для виведення результатів послідовних кроків при знаходженні похідної; локальні процедури **`пароль_файл`** та **`пароль_клавіатура`** – для зчитування відповідно пароля із задалегідь підготовленого файла за указаною в програмі адресою та пароля, що вводить користувач з клавіатури.; процедура **s_diff**, що експортується – для перевірки прав користувача на доступ до процедур модуля та передачі цим процедурам виразу, який потрібно продиференціювати.

Приведемо тексти указаних процедур:

```
> d_pow := proc( f, x )
  local rest;
  if not has( f, x) then 0
  elif f=x then 1
  elif type( f, `+` ) then map(`procname`, f, x )
  elif type( f, `*` ) then
    rest := mul(op(i,f), i=2..nops(f));
    `procname`(op(1,f), x)*rest + op(1,f)*`procname`(rest, x);
  elif type( f, `^` ) then
    if has(op(1,f), x) and not has(op(2,f), x) then
      op(2,ff)*op(1,ff)^(op(2,ff)-1)*`procname`(op(1,ff), xx)
    else
      ['d'( f )/d||x]
    end if
  else
  end if
end if
```

```

    ['d'( f )/d||x]
end if
end proc;

```

```

show_dif := proc( ff, xx )
    local kof,rest,rest1, i;
    if not has( ff, xx ) then
        print(d(ff)/d||xx=0)
    elif ff=xx then print(d(ff)/d||xx=1)
    elif type( ff, `+` ) then
        print(d(ff)/d||xx=add(d(op(i,ff))/d||xx,i=1..nops(ff))):
        for i to nops(ff) do
            `procname`(op(i,ff), xx)
        end do;
    elif type( ff, `*` ) then
        rest1:=remove(has,ff,xx);
    ... ..
    elif type( ff, `^` ) then
        if has(op(1,ff), xx) and not has(op(2,ff), xx) then
            print(d(ff)/d||xx=op(2,ff)*op(1,ff)^(op(2,ff)-
                1)*`if`(op(1,ff)=xx,1,[d(op(1,ff))/d||xx]))
        else
            'd'( ff )/d||xx
        end if
    else
        'd'( ff )/d||xx
    end if;
d(ff)/d||xx=d_poll1(ff,xx)
end proc;

```

```

`пароль_файл`:=proc()
local p_f,fo;
try
fo:=fopen("C:\\System Volume Information\\my_data.txt",READ,TEXT):
p_f:=readline(fo);
catch "file or directory does not exist": RETURN(`Д О С Т У П    З А Б О
P О Н Е Н О!!!!`);
end try;
fclose(fo):
p_f
end proc;

```

```

`пароль_клавіатура`:=proc()
local `Пароль_для_доступа`;
`Пароль_для_доступа`:=readstat("Уведіть пароль: ");
convert(`Пароль_для_доступа`,string);
end proc;

```

```

`s_diff`:=proc(expr, var)
local `p_file`,`p_t`;
`p_file`:=`пароль_файл`() ;
if type(`p_file`,symbol) then
    `p_file`
else
    `пароль_користувача`:=`if`(not
assigned(`пароль_користувача`),`пароль_клавіатура`() ,`пароль_користувача`);
    if `пароль_користувача`=`p_file` then

```

```

Passvord_Tr_F:=true:
`show_dif`(expr, var)
else
if not assigned(Passvord_Tr_F) then
`пароль_користувача`:=`пароль_користувача`;
RETURN(printf("%s\n", `П А Р О Л Ь      Н Е В І Р Н И Й !`))
else
RETURN(printf("%s\n", `Д О С Т У П      З А К Р И Т О !`));
end if
end if
end if
end proc;

```

З метою економії місця текст процедури `show_dif` приведено не повністю.

Приведені процедури `show_dif` та `d_pow` уміють диференціювати степеневу функцію x^n та „знають” правила диференціювання суми та добутку функцій. Звичайно, ці процедури можна удосконалити для можливості диференціювання довільних виразів.

Процедуру перевірки `s_dif` побудовано так, що при кожному звертанні до процедур модуля відбувається зчитування пароля з файла "C:\\System Volume Information\\my_data.txt" та порівняння його з паролем, що введений користувачем. Запит користувачу на введення пароля здійснюється тільки при першому звертанні до процедур модуля. Пароль, уведений користувачем, зберігається під час поточного сеансу роботи з даним модулем. Така структура дозволяє регулювати доступ користувача вилученням - записом файла "C:\\System Volume Information\\my_data.txt" або зміною його змісту.

Блок `try - catch - end try` в процедурі `пароль_файл`` призначений для обробки виняткового стану, який виникає за умови відсутності файла "C:\\System Volume Information\\my_data.txt".

Для користування модулем зручно його додати до стандартної бібліотеки Maple:

```
> savelib('_Dif');
```

опція модуля - `package` забезпечила збереження його у вигляді пакета тільки з однією доступною командою `s_diff`. Звертатись до цієї команди тепер можна як до будь-якої іншої команди певного пакета:

```
> _Dif:-s_diff(3*z^2+sin(x)*z, z);
```

Уведіть пароль: ПОХІДНА;

$$\frac{d(3z^2 + \sin(x)z)}{dz} = \frac{d(3z^2)}{dz} + \frac{d(\sin(x)z)}{dz}$$

$$\frac{d(3z^2)}{dz} = \{3\} \left[\frac{d(z^2)}{dz} \right]$$

$$\frac{d(z^2)}{dz} = 2z$$

$$\frac{d(\sin(x)z)}{dz} = \sin(x)$$

$$\frac{d(3z^2 + \sin(x)z)}{dz} = 6z + \sin(x)$$

Якщо підключити пакет командою:

```
> with(_Dif):
```


то синтаксис звернення до команди `s_diff` спрощується – ім'я пакета указувати не потрібно:

```
> s_diff(3/z+ln(x)^2*z^(2/3), z);
```

$$\frac{d\left(\frac{3}{z} + \ln(x)^2 z^{(2/3)}\right)}{dz} = \frac{d\left(\frac{3}{z}\right)}{dz} + \frac{d(\ln(x)^2 z^{(2/3)})}{dz}$$

$$\frac{d\left(\frac{3}{z}\right)}{dz} = \{3\} \left[\frac{d\left(\frac{1}{z}\right)}{dz} \right]$$

$$\frac{d\left(\frac{1}{z}\right)}{dz} = -\frac{1}{z^2}$$

$$\frac{d(\ln(x)^2 z^{(2/3)})}{dz} = \{\ln(x)^2\} \left[\frac{d(z^{(2/3)})}{dz} \right]$$

$$\frac{d(z^{(2/3)})}{dz} = \frac{2}{3 z^{(1/3)}}$$

$$\frac{d\left(\frac{3}{z} + \ln(x)^2 z^{(2/3)}\right)}{dz} = -\frac{3}{z^2} + \frac{2 \ln(x)^2}{3 z^{(1/3)}}$$

Спробуємо переглянути програмний код модуля та його процедур. Установимо потрібне значення змінної інтерфейсу `verboseproc`

```
> interface(verboseproc=2) :
> print(f);
```

show_dif

Це означає, що всі локальні процедури модуля – недоступні для перегляду та копіювання. Вилучимо файл "C:\\System Volume Information\\my_data.txt":

```
> s_diff(3/z+ln(x)^2*z^(2/3), z);
```

ДОСТУП ЗАБОРОНЕНО!!!

Аналогічне повідомлення з'являється і при зміні змісту указанного файла. При відновленні файла робота з командою `s_diff` може бути продовжена.

Звичайно, збереження пароля в незашифрованому вигляді у текстовому файлі `my_data.txt` робить систему захисту нестійкою до її зламу. Один із найпростіших шляхів підвищення стійкості системи захисту є, наприклад, збереження пароля в файлі спеціального Maple-формату з розширенням `m`; збереження пароля у зашифрованому вигляді та ін. Але це уже питання підвищення рівня захисту, які практично невичерпні.

Висновки

Конструкція `module` не тільки дозволяє розробити ефективний захист програмного коду процедур користувача від несанкціонованого перегляду та копіювання але й побудувати зручний механізм регулювання доступом до роботи з цими процедурами.

Список літератури

1. Михалевич В.М. Реалізації технології "живих сторінок" в Maple, MathCad, Excel // Вісник ВПІ. – 2004. - № 3. – С. 90-95.
2. Mikhalevich V. "Development of Electronic Courses in Maple Environment on a Process Engineering of a "Live Pages". // "Інтернет – Освіта - Наука - 2004", четверта міжнародна конференція ІОН – 2004, 28 вересня – 16 жовтня, 2004 р. Збірник матеріалів конференції. Том 1. – Вінниця: Універсум-Вінниця, 2004. - С.31-34.
3. Михалевич В.М. Навчально-контролюючий Maple – комплекс з вищої математики //Інформаційні технології та комп'ютерна інженерія. – 2004. - № 1. – С.74-78.
4. Михалевич В.М. Excel-VBA-Maple програма генерації задач з дисциплін математичного спрямування//Інформаційні технології та комп'ютерна інженерія. – 2005. - № 2. – С.74-83.
5. Михалевич В. М., Крупський Я. В. Аналіз сучасного стану питань генерування завдань з вищої математики. // "Інтернет – Освіта - Наука - 2006", п'ята міжнародна конференція ІОН – 2006, 10-14 жовтня, 2006 р. Збірник матеріалів конференції. Том 1. – Вінниця: УНІВЕРСУМ-Вінниця, 2006. - С.31-34.
6. Михалевич В.М. Ключові проблеми створення навчально-контролюючого комплексу з дисциплін математичного спрямування// Сучасні інформаційні технології та іноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми// Зб. наук. прац. – Випуск 10 / Редкол.: І.А. Зязюн (голова) та ін. Київ-Вінниця: ДОВ "Вінниця", 2006, С.391-197.
7. Михалевич В.М., Шевчук О.І., Буга Н.Л. Сучасні інформаційні технології та іноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми// Зб. наук. прац. – Випуск 14 / Редкол.: І.А. Зязюн (голова) та ін. Київ-Вінниця: ДОВ "Вінниця", 2007, С.357-360.
8. Матросов А.В. Maple 6. Решение задач высшей математики и механики. – СПб.: БХВ-Петербург, 2001. – 528 с.
9. Васильев А.Н. Maple 8. Самоучитель.: - М.: Издательский дом «Вильямс», 2003. – 353 с.
10. Аладьев В.З. Эффективная работа в Maple 6/7 – М.: Лаборатория Базовых Знаний, 2002. – 336 с.
11. Дьяконов В.П. Maple 7: учебный курс. – СПб.: Питер, 2002. – 672 с.
12. Манзон Б.М. Maple V Power Edition – М.: Информационно-издательский дом «Филин», 1998. – 240 с.
13. Maple 9/ Introductory Programming Guide/M.V. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, J. McCarron, P. DeMarco. Canada. Maplesoft, division of Waterloo Maple Inc. 2003. 388 p.
14. Maple 9/ Advanced Programming Guide/M.V. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, J. McCarron, P. DeMarco. Canada. Maplesoft, division of Waterloo Maple Inc. 2003. 444 p.

Михалевич Володимир Маркусович – зав. кафедри прикладної математики ВНТУ, доктор технічних наук, професор, тел. (8-0432)-598-594. e-mail: pmatmih@vstu.vinnica.ua

Димніч Ігор Віталійович – керівник відділу розробки фірми DID World Wide™, Ізраїль.

Михалевич Олексій Володимирович – студент гр. ІБС-07, ІНІТКІ, ВНТУ

Показано, що управлінська діяльність може бути описана оператором у побудованому інформаційному просторі. Показано, що управлінська діяльність може бути представлена у вигляді сукупності двохкомпонентних операторів спеціального вигляду. Доведено, що мінімально можливий клас типів гравців становить 16 різних класів стратегій діяльності, які є універсальними і які окремий тип проявляє завжди. Таким чином, побудована повна типологія гравців: тип тут розуміється як «тип стратегій здійснення діяльності».

УДК 004.05

Оптимізація процесу пошуку заповідів відмовлення РЕА в умовах дефіциту інформації / О.Л.Овчинніков, Г.Г. Сергеев // Інформаційні технології та комп'ютерна інженерія.–2007.–№3(10). – С. 140-143: іл. 1. Табл. 0. Бібліогр.: 4 назв.

Розглянуто задачу оптимізації процесу пошуку заповідів відмовлення радіоелектронної апаратури, у випадку неповної схемної документації виробу. Формалізовано критерії оцінки ефективності процесу діагностики і ремонту РЕА. Запропоновано алгоритм, що дозволяє зменшити кількість операцій внутрішнього тестування, необхідних для пошуку несправного елемента.

УДК 621.372.8

Аналіз інформаційних технологій моніторингу та контролю оптичних волокон / С.В. Павлов, Р.В.Просоловський // Інформаційні технології та комп'ютерна інженерія.–2007.–№3(10). – С. 144-154: іл. 8. Табл. 1. Бібліогр.: 10 назв.

У статті авторами проведено системний аналіз інформаційних методів та технологій для моніторингу та контролю оптичних волокон, а також їх порівняння у контексті вимірювання параметрів оптичних волокон, приведено їх переваги та недоліки, розглянуто похибки методів вимірювання затухань.

УДК 381.3.06

Модульне проектування оптимальної системи захисту інформаційних ресурсів / А.В. Дудат'єв // Інформаційні технології та комп'ютерна інженерія.–2007.–№3(10). – С. 155-158: іл. 1. Табл. 0. Бібліогр.: 5 назв.

В даній статті розглянуто та обґрунтовано доцільність використання модульного методу проектування системи захисту інформаційних ресурсів. Запропонована математична модель, яка дозволяє отримати оптимальну систему захисту інформаційних ресурсів за показниками різних параметрів можливих механізмів захисту.

УДК 681.3.06

Захист Maple-процедур / В. М. Михалевича, І.В. Димніча, О. В. Михалевича // Інформаційні технології та комп'ютерна інженерія.–2007.–№3(10). – С. 159-165: іл. 0. Табл. 0. Бібліогр.: 14 назв.

В статті запропоновано технологію захисту Maple – процедур від несанкціонованого доступу та перегляду. Запропонований механізм захисту процедур розроблено на основі конструкції module. Наведено приклад, який демонструє, що розроблена технологія не тільки дозволяє створити ефективний захист програмного коду процедур користувача від несанкціонованого перегляду та копіювання, але й побудувати зручний механізм регулювання доступом до роботи з цими процедурами.

УДК 004.7:004.056

Метод оцінки ступеня організованості систем захисту інформації / Н.И. Алішов, И.А. Жуков, В.А.Марченко // Інформаційні технології та комп'ютерна інженерія.–2007.–№3(10). – С. 166-171: іл. 1. Табл. 2. Бібліогр.: 16 назв.

У статті даний короткий аналіз стандартів оцінювання захищеності інформації а також областей їхнього застосування. Описано методику оцінки ступеня організованості систем захисту інформації, що дозволяє проводити оцінку не тільки на етапі експлуатації, але й на етапі проектування. Наведено математичний опис методу оцінювання, а також дані деякі практичні рекомендації з його застосування.

УДК 004.934

Транскодування стиснених мовних сигналів між GSM 06.20 та G.729A / Р.П. Шевчук // Інформаційні технології та комп'ютерна інженерія.–2007.–№3(10). – С. 172-177: іл. 3. Табл. 3. Бібліогр.: 11 назв.

У роботі запропоновано метод транскодування стиснених мовних сигналів між G.729A та GSM 06.20. Запропонований метод враховує структурну схожість модулів короткотермінової фільтрації, довготермінової фільтрації та випадкового збудження алгоритмів CS-ACELP та VSELP, яка дозволяє

активов, а также дополнение ими классических линейных методов моделирования. В ходе исследования были построены регрессионная модель процесса прогнозирования и оптимизированная модель с введенной GARCH-погрешностью. Результаты использования разработанных моделей представлены в программной среде MathCad, и свидетельствуют о необходимости оптимизации та дополнения традиционных линейных методов прогнозирования поведения финансовых активов нелинейными.

УДК 519

Информационное пространство и классификация стратегий управленческой деятельности в теории игр и принятия решений / А.А. Шиян // Информационные технологии и компьютерная инженерия.–2007.–№3(10). – С. 131-139: ил. 0. Табл. 1. Библиогр.: 20 назв.

Показано, что управленческая деятельность может быть описана оператором в построенном информационном пространстве. Показано, что управленческая деятельность может быть представлена в виде совокупности двухкомпонентных операторов специального вида. Доказано, что минимально возможный класс типов игроков составляет 16 различных классов стратегий деятельности, которые являются универсальными и которые отдельный тип проявляет всегда. Таким образом, построена полная типология игроков: тип здесь понимается как «тип стратегий осуществления деятельности».

УДК 004.05

Оптимизация процесса поиска причины отказа РЭА в условиях дефицита информации / А.Л.Овчинников, Г.Г. Сергеев / Информационные технологии и компьютерная инженерия.–2007.–№3(10). – С. 140-143: ил. 1. Табл. 0. Библиогр.: 4 назв.

Рассмотрена задача оптимизации процесса поиска причины отказа радиоэлектронной аппаратуры, в случае неполной схемной документации изделия. Формализованы критерии оценки эффективности процесса диагностики и ремонта РЭА. Предложен алгоритм, позволяющий уменьшить количество операций внутрисхемного тестирования, необходимых для поиска неисправного элемента.

УДК 621.372.8

Анализ информационных технологий мониторинга и контроля оптических волокон / С.В. Павлов, Р.В. Просоловский // Информационные технологии и компьютерная инженерия.–2007.–№3(10). – С. 144-154:ил. 8. Табл. 1. Библиогр.: 10 назв.

В статье авторами проведен системный анализ информационных методов и технологий для мониторинга и контроля оптических волокон, а также их сравнение в контексте измерения параметров оптических волокон, проанализированы их преимущества и недостатки, рассмотрены погрешности методов измерения затуханий.

УДК 381.3.06

Модульное проектирование оптимальной системы защиты информационных ресурсов / А.В.Дудатьев // Информационные технологии и компьютерная инженерия.–2007.–№3(10). – С. 155-158: ил. 1. Табл. 0. Библиогр.: 5 назв.

В данной статье рассмотрена и обоснована целесообразность использования модульного метода проектирования системы защиты информационных ресурсов. Предложена математическая модель, которая позволит получить оптимальную систему защиты информационных ресурсов по показателям разных параметров возможных механизмов защиты.

УДК 681.3.06

Защита Maple-процедур / В. М. Михалевица, И.В. Димніча, О. В. Михалевица // Информационные технологии и компьютерная инженерия.–2007.–№3(10). – С. 159-165: ил. 0. Табл. 0. Библиогр.: 14 назв.

В статье предложена технология защиты Maple - процедур от несанкционированного доступа и просмотра. Предложенный механизм защиты процедуры разработан на основе конструкции module. Приведен пример, который демонстрирует, что разработанная технология не только позволяет создать эффективную защиту программного кода процедур пользователя от несанкционированного просмотра и копирования, но и построить удобный механизм регулирования доступом к работе с этими процедурами.

УДК 004.7:004.056

Метод оценки степени организованности систем защиты информации / Н.И. Алишов, И.А. Жуков, В.А. Марченко // Информационные технологии и компьютерная инженерия.–2007.–№3(10). – С. 166-171: ил. 1. Табл. 2. Библиогр.: 16 назв.

В статье дан краткий анализ стандартов оценивания защищенности информации, а также областей их применения. Описана методика оценки степени организованности систем защиты