(19) UA (11) 77 187 (13) C2

(51)МПК

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ГОСУДАРСТВЕННЫЙ ДЕПАРТАМЕНТ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ УКРАИНЫ

(21), (22) Заявка: 2004010473, 04.07.2002

(24) Дата начала действия патента: 15.11.2006

(30) Приоритет: 31.07.2001 FR 01/10244

(46) Дата публикации: 15.11.2006G06F 21/00 20060101CFI20061012RMUA G06F 1/00 20060101CLI20061012RMUA

(86) Заявка PCT:
PCT/FR02/02343, 20020704

(72) Изобретатель:
Кено Жан-Кристоф, FR,
Сгро Жиль, FR

(73) Патентовладелец:
ВАЛИДИ, FR

(54) СПОСОБ ЗАЩИТЫ ПРОГРАММЫ ОТ НЕРАЗРЕШЕННОГО ДОСТУПА С ПОМОЩЬЮ КОНТРОЛЯ СООТВЕТСТВИЯ ЗАДАННОМУ КРИТЕРИЮ

(57) Реферат:

Настоящее изобретение относится к способу защиты, по меньшей мере, одной программы из состава программного обеспечения системы обработки данных от неразрешенного доступа. Предлагаемый способ заключается в том, что выбирают в программном обеспечении характеристику, контролируемую, по меньшей мере, при выполнении программы, и критерий, которому должна соответствовать указанная характеристика, определяют элементы программы, позволяющие обнаружить несоответствие характеристики заданному критерию, и элементы, обеспечивающие передачу сообщения о несоответствии в систему обработки данных или изменяющие алгоритм выполнения программы таким образом, чтобы обеспечить указанное соответствие.

Официальный бюлетень "Промышленная собственность". Книга 1 "Изобретения, полезные модели, топографии интегральных микросхем", 2006, N 11, 15.11.2006. Государственный департамент интеллектуальной собственности Министерства образования и науки Украины.

UA 77187 C2

UA 77187 C2

(19) UA (11) 77 187 (13) C2

(51) Int. Cl.

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

STATE DEPARTMENT OF INTELLECTUAL PROPERTY

(12) DESCRIPTION OF PATENT OF UKRAINE FOR INVENTION

(21), (22) Application: 2004010473, 04.07.2002

(24) Effective date for property rights: 15.11.2006

(30) Priority: 31.07.2001 FR 01/10244

(46) Publication date: 15.11.2006G06F 21/00 20060101CFI20061012RMUA G06F 1/00 20060101CLI20061012RMUA

(86) PCT application:
PCT/FR02/02343, 20020704

(72) Inventor:
Keno Jean-Christof, FR,
Sgro Jyle, FR

(73) Proprietor:
VALIDY, FR

(54) METHOD FOR PROTECTING SOFTWARE AGAINST UNAUTHORIZED USE BY CONTROLLING THE CORRESPONDENCE WITH A SPECIFIED CRITERION

(57) Abstract:

The invention concerns a method for protecting, from a unit, a vulnerable software against its unauthorised use, said vulnerable software operating on a data processing system. The method consists in defining: at least one executing characteristic of the software, capable of being monitored at least partly in a unit; at least one criterion to be observed for at least one executing characteristic of the software; detection means to be implemented in a unit and enabling to detect whether at least one software executing characteristic does not observe at least one associated criterion; and coercion means to be implemented in a unit for informing the data processing system and/or modifying the execution of a software when at least one criterion is not observed.

Official bulletin "Industrial property". Book 1 "Inventions, utility models, topographies of integrated circuits", 2006, N 11, 15.11.2006. State Department of Intellectual Property of the Ministry of Education and Science of Ukraine.

UA 77187 C2

UA 77187 C2

(19) **UA** (11) **77 187** (13) **C2**

(51)МПК

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ДЕПАРТАМЕНТ
ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ

(12) ОПИС ВИНАХОДУ ДО ПАТЕНТУ УКРАЇНИ

(21), (22) Дані стосовно заявки:
2004010473, 04.07.2002

(24) Дата набуття чинності: 15.11.2006

(30) Дані стосовно пріоритету відповідно до Паризької конвенції : 31.07.2001 FR 01/10244

(46) Публікація відомостей про видачу патенту (деклараційного патенту): 15.11.2006G06F 21/00 20060101CFI20061012RMUA G06F 1/00 20060101CLI20061012RMUA

(86) Номер та дата подання міжнародної заявки відповідно до договору PCT:
PCT/FR02/02343, 20020704

(72) Винахідник(и):
Кено Жан-Крістоф , FR,
Сгро Жиль , FR

(73) Власник(и):
ВАЛІДІ, FR

(54) СПОСІБ ЗАХИСТУ ПРОГРАМИ ВІД ЇЇ НЕАВТОРИЗОВАНОГО ВИКОРИСТАННЯ ЗА ДОПОМОГОЮ ТАК ЗВАНОГО ПРИНЦИПУ ДЕТЕКТУВАННЯ Й ПРИМУСУ

(57) Реферат:

Спосіб захисту програми від її неавторизованого використання за допомогою так званого принципу детектування й примусу належить до захисту уразливих програм з використанням щонайменше одного пристрою захисту. Спосіб полягає в тому, щоб визначити: щонайменше одну характеристику виконання програми, яка може бути проконтрольована в певному пристрої; щонайменше один критерій, який має виконуватися щонайменше для однієї характеристики виконання програми; засоби детектування, що їх слід застосувати у пристрої і які дозволяють виявити, що принаймні одна характеристика виконання програми не задовольняє принаймні одним відповідний критерій.

UA 77187 C2

UA 77187 C2

Опис винаходу

Даний винахід належить до галузі систем обробки даних у широкому сенсі цього слова, а точніше кажучи, винахід відноситься до засобів захисту від неавторизованого використання програми, що функціонує в зазначених системах обробки даних.

Предмет даного винаходу відноситься, конкретніше, до способів захисту програм від неавторизованого використання на основі пристрою обробки і зберігання, причому цей пристрій звичайно реалізується у вигляді карти з мікрочипом або апаратного ключа на порту USB.

У згаданій галузі техніки основна проблема пов'язана з неавторизованим використанням програм користувачами, які не набули права на їхнє використання (ліцензію). Таке незаконне використання програм завдає явного збитку виробникам і продавцям програм, і/або будь-яким іншим особам, що використовують такі програми у своїх виробах. Щоб уникнути створення таких незаконних копій, для захисту подібних програм було запропоновано ряд рішень на основі наявних технічних можливостей.

Так, відомий спосіб захисту, що полягає у використанні апаратного захисного елемента - фізичного пристрою, названого електронним захисним ключем-заглушкою ("dongle"). Такий захисний ключ має гарантувати виконання програми тільки за наявності ключа. Треба, проте, констатувати, що це вирішення неефективне, бо його можна легко обійти. Зловмисник може за допомогою спеціальних прийомів, наприклад дизасемблювання, вилучити команди контролю захисного ключа. При цьому стає можливим виготовити незаконні копії модифікованих версій програми, що не мають жодного захисту. Крім того, це вирішення не можна поширити на всі програми з огляду на складність підключення більш ніж двох захисних ключів до однієї системи.

Відповідно, задача, розв'язувана даним винаходом, полягає в усуненні зазначених хиб шляхом створення способу захисту програми від неавторизованого доступу, що використовує пристрій обробки і зберігання, спеціально створений для цієї мети, в тій мірі, у якій наявність такого пристрою необхідна для повноцінної роботи програми.

Вирішення даної задачі досягнуто згідно з винаходом створенням способу захисту (на основі щонайменше одного незадіяного пристрою, що містить щонайменше засоби обробки і засоби запам'ятовування) від неавторизованого використання вразливої програми, яка функціонує в системі обробки даних. Спосіб згідно з винаходом полягає в тому, що:

- у фазі захисту:

- визначають:

- щонайменше одну характеристику виконання програми, що може бути проконтрольована, щонайменше частково, у пристрої,

- щонайменше один критерій, що має виконуватися щонайменше для однієї характеристики виконання програми,

- засоби детектування, що їх слід застосовувати у пристрої і які дозволяють виявити, що принаймні одна характеристика виконання програми не відповідає щонайменше одному відповідному критерію,

- і засоби примусу, що їх слід застосовувати у пристрої та які дозволяють проінформувати систему обробки даних і/або модифікувати виконання програми, поки не дотримано хоча б одного критерію,

- конструюють засоби експлуатації, що дозволяють пристрою задіяти засоби детектування й засоби примусу;

створюють захищену програму:

- за допомогою вибору щонайменше однієї характеристики виконання контрольованої програми серед характеристик виконання, які можуть бути проконтрольовані,

- за допомогою вибору щонайменше одного критерію, який має виконуватися щонайменше для однієї вибраної характеристики виконання програми,

- за допомогою вибору щонайменше одного алгоритму, який у ході виконання вразливої програми використовує щонайменше один операнд і дозволяє одержати щонайменше один результат і для якого слід контролювати щонайменше одну характеристику виконання вибраної програми,

- за допомогою вибору щонайменше одного фрагмента коду вразливої програми, що містить щонайменше один вибраний алгоритм,

- за допомогою створення вихідного коду захищеної програми на основі коду вразливої програми модифікацією щонайменше одного вибраного фрагмента коду вразливої програми, щоб одержати щонайменше один модифікований фрагмент коду захищеної програми, причому ця модифікація така, що:

- в ході виконання захищеної програми перша виконувана частина виконується в системі обробки даних, а друга виконувана частина виконується у пристрої, який також містить засоби обробки, отриманому з незадіяного пристрою після завантаження інформації,

- друга виконувана частина виконує щонайменше функціональну можливість щонайменше одного вибраного алгоритму,

- і в ході виконання захищеної програми щонайменше одна вибрана характеристика виконання контролюється за допомогою другої виконуваної частини, і недотримання критерію приводить до модифікації виконання захищеної програми;

- і за допомогою створення:

- першої частини об'єктного коду захищеної програми, причому перша частина об'єктного коду така, що в ході виконання захищеної програми реалізується перша виконувана частина, яка виконується в системі обробки даних, і щонайменше частина якої враховує, що контролюється принаймні одна характеристика виконання

UA 77187 C2

вибраної програми,

- і другої частини об'єктного коду захищеної програми, що містить засоби експлуатації, що дозволяють застосовувати також засоби детектування й засоби примусу, причому друга частина об'єктного коду така, що після завантаження в незадіяний пристрій, у ході виконання захищеної програми реалізується друга виконувана частина, за допомогою якої контролюється принаймні одна характеристика виконання програми і за допомогою якої недотримання критерію приводить до модифікації виконання захищеної програми.

- і завантажують другу частину об'єктного коду в незадіяний пристрій з одержанням пристрою,

- а у фазі (U) використання, в ході якої відбувається виконання захищеної програми:

- в присутності пристрою:

- якщо всіх критеріїв, що відповідають усім контрольованим характеристикам виконання всіх модифікованих фрагментів захищеної програми, дотримано, припускають номінальне функціонування зазначених фрагментів захищеної програми й, отже, номінальне функціонування захищеної програми,

- а якщо принаймні одного з критеріїв, що відповідають характеристиці контрольованого виконання одного фрагмента захищеної програми, не дотримано, інформують систему обробки даних про зазначене недотримання і/або модифікують функціонування фрагмента захищеної програми таким чином, щоб функціонування захищеної програми було змінено.

- тоді як за відсутності пристрою, незважаючи на запит фрагмента першої виконуваної частини на запуск виконання у пристрої функціональної можливості вибраного алгоритму, не забезпечується можливість коректно відповісти на цей запит, так що принаймні згаданий фрагмент не виконується коректно й, отже, захищена програма не є повнофункціональною.

Відповідно до варіанта здійснення способу за винаходом:

- у фазі захисту:

- визначають:

- як характеристику виконання програми, яка може бути проконтрольована, - змінну для кількісного контролю використання однієї функціональної можливості програми,

- як критерій, що його слід дотримуватися, - щонайменше одне порогове значення, зв'язане з кожною змінною для кількісного контролю,

- і засоби поновлення, що дозволяють поновити щонайменше одну змінну для кількісного контролю;

- конструюють засоби експлуатації, що дозволяють пристрою застосовувати засоби поновлення;

- і модифікують захищену програму:

- за допомогою вибору як характеристики виконання контрольованої програми щонайменше однієї змінної для кількісного контролю використання однієї функціональної можливості програми,

- за допомогою вибору:

- щонайменше однієї функціональної можливості захищеної програми, використання якої можна проконтролювати з використанням змінної для кількісного контролю,

- щонайменше однієї змінної для кількісного контролю, яка служить для кількісної характеристики використання згаданої функціональної можливості,

- щонайменше одного порогового значення, пов'язаного з вибраною змінною для кількісного контролю і відповідного межі використання зазначеної функціональної можливості,

- і щонайменше одного методу поновлення значення згаданої змінної для кількісного контролю залежно від використання зазначеної функціональної можливості,

- і за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми, причому зазначена модифікація така, що в ході виконання захищеної програми змінна для кількісного контролю поновлюється за допомогою другої виконуваної частини, залежно від використання зазначеної функціональної можливості, і враховується щонайменше одне перевищення порогового значення,

- а у фазі використання, в присутності пристрою, в разі, якщо виявлено щонайменше одне перевищення порогового значення, що відповідає щонайменше одній межі використання, інформують про це систему обробки даних і/або модифікують функціонування фрагмента захищеної програми, таким чином, щоб функціонування захищеної програми було змінено.

Відповідно до іншого варіанта здійснення способу згідно з винаходом:

- у фазі захисту:

- визначають:

- декілька відповідних порогових значень щонайменше для однієї змінної для кількісного контролю,

- і різноманітні засоби примусу, що відповідають кожному зі згаданих порогів;

- і модифікують захищену програму:

- за допомогою вибору у вихідному коді захищеної програми щонайменше однієї змінної для кількісного контролю, з якою мають зв'язуватися декілька порогових значень, що відповідають різноманітним межам використання функціональної можливості,

- за допомогою вибору щонайменше двох порогових значень, зв'язаних з вибраною змінною для кількісного контролю,

- і за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми, причому зазначена модифікація така, що в ході виконання захищеної програми перевищення різноманітних порогових значень враховуються за допомогою другої виконуваної частини різноманітними способами,

- а у фазі використання:

- в присутності пристрою:

UA 77187 C2

UA 77187 C2

- у випадку, коли виявлене перевищення першого порогового значення, дають команду захищеній програмі не використовувати надалі відповідну функціональну можливість,

- а в разі, коли виявлене перевищення другого порогового значення, роблять нездійсненною відповідну функціональну можливість і/або щонайменше частину захищеної програми.

Відповідно до ще одного варіанта здійснення способу згідно з винаходом:

- в фазі захисту:

- визначають засоби перезавантаження, що дозволяють щонайменше одне додаткове використання щонайменше однієї функціональної можливості програми, контрольованої за допомогою змінної для кількісного контролю;

- конструюють засоби експлуатації, що дозволяють також пристрою задіяти засоби перезавантаження;

- і модифікують захищену програму:

- за допомогою вибору у вихідному коді захищеної програми щонайменше однієї змінної для кількісного контролю, що дозволяє обмежити використання однієї функціональної можливості, для якої існує можливість дозволу щонайменше на одне додаткове використання,

- і за допомогою модифікації щонайменше одного вибраного фрагмента, причому зазначена модифікація така, що у фазі перезавантаження щонайменше одне додаткове використання щонайменше однієї функціональної можливості, що відповідає одній вибраній змінній для кількісного контролю, може бути дозволено,

- а у фазі перезавантаження:

- поновлюють щонайменше одну вибрану змінну для кількісного контролю і щонайменше одне відповідне порогове значення так, щоб дозволити щонайменше одне додаткове використання функціональної можливості.

Відповідно до варіанта здійснення способу згідно з винаходом:

- у фазі захисту:

- визначають:

- як характеристику виконання програми, яка може бути проконтрольована, - профіль використання програми,

- і як критерій, що його слід дотримуватися, - щонайменше одну ознаку виконання програми;

- і модифікують захищену програму:

- за допомогою вибору як характеристики виконання контрольованої програми щонайменше одного профілю використання програми,

- за допомогою вибору щонайменше однієї ознаки виконання програми, що її має дотримуватися щонайменше один профіль використання,

- і за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми, причому зазначена модифікація така, що в ході виконання захищеної програми друга виконувана частина дотримується всіх вибраних ознак виконання,

- а у фазі використання, в присутності пристрою в разі, якщо виявлено, що не дотримано хоча б однієї ознаки виконання, інформують про це систему обробки даних і/або модифікують функціонування частини захищеної програми так, щоб функціонування захищеної програми було змінено.

Відповідно до подальшого варіанта здійснення способу згідно з винаходом:

- в фазі захисту:

- визначають:

- набір інструкцій, інструкції зі складу якого можуть виконуватися у пристрої,

- набір команд інструкцій для згаданого набору інструкцій, причому команди інструкцій можуть бути виконані в системі обробки даних, викликаючи у пристрої виконання інструкцій,

- як профіль використання - зчеплення інструкцій,

- як ознаку виконання - бажане зчеплення для виконання інструкцій,

- як засоби детектування - засоби, що дозволяють виявити, що зчеплення інструкцій не відповідає бажаному,

- як засоби примусу - засоби, що дозволяють проінформувати систему обробки даних і/або модифікувати функціонування фрагмента захищеної програми, якщо зчеплення інструкцій не відповідає бажаному;

- конструюють засоби експлуатації, що дозволяють пристрою виконувати інструкції з набору інструкцій, причому виконання згаданих інструкцій викликається виконанням команд інструкцій у системі обробки даних;

- і модифікують захищену програму:

- за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми:

- за допомогою перетворення елементарних функцій на інструкції,

- за допомогою завдання зчеплення, що його мають дотримуватися щонайменше деякі з інструкцій під час їхнього виконання у пристрої,

- і за допомогою перетворення елементарних команд на команди інструкцій, що відповідають використовуваним інструкціям,

- а у фазі використання, в присутності пристрою, в разі, якщо виявлено, що зчеплення виконуваних у пристрої інструкцій не відповідає бажаному, інформують про це систему обробки даних і/або модифікують функціонування фрагмента захищеної програми так, щоб функціонування захищеної програми було змінено.

Відповідно до іншої кращої форми реалізації способу згідно з винаходом:

- у фазі захисту:

- визначають:

- як набір інструкцій - набір інструкцій, у якому щонайменше деякі інструкції працюють на регістрах і використовують щонайменше один операнд для видачі результату,

UA 77187 C2

UA 77187 C2

- щонайменше для частини інструкцій, що працюють на регістрах:
- частину, що задає функціональну можливість інструкції,
- і частину, що задає бажане зчеплення для виконання інструкцій і містить бітові поля, які відповідають:
- полю ідентифікації інструкції,
- для кожного операнда інструкції:
- полю прапора,
- і полю ідентифікації, передбаченої для операнда,
- для кожного регістра, що належить до засобів експлуатації і використовується набором інструкцій, - поле передбаченої ідентифікації, в якому автоматично запам'ятовується ідентифікація останньої інструкції, що повернула свій результат у вказаний регістр,
- як засоби детектування - засоби, що дозволяють під час виконання інструкції для кожного операнда, коли цього вимагає поле прапора, контролювати рівність між полем генерованої ідентифікації, що відповідає регістру, використовуваному вказаним операндом, і полем передбаченої ідентифікації початкової адреси цього операнда,
- як засоби примусу - засоби, що дозволяють модифікувати результат виконання інструкцій, якщо принаймні одна з контрольованих рівностей хибна.

Відповідно до кращої форми реалізації способу згідно з винаходом:
- у фазі захисту:
- модифікують захищену програму:
- за допомогою вибору щонайменше однієї змінної, використовуваної щонайменше в одному вибраному алгоритмі, яка в ході виконання захищеної програми частково визначає стан захищеної програми,
- за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми щонайменше одна вибрана змінна або одна копія вибраної змінної знаходиться у пристрої,
- і за допомогою створення:
- першої частини об'єктного коду захищеної програми, причому ця перша частина об'єктного коду така, що в ході виконання захищеної програми щонайменше один фрагмент першої виконуваної частини враховує, що принаймні одна змінна або щонайменше одна копія змінної знаходиться у пристрої,
- і другої частини об'єктного коду захищеної програми, причому ця друга частина об'єктного коду така, що після завантаження у пристрій і в ході виконання захищеної програми з'являється друга виконувана частина, за допомогою якої щонайменше одна вибрана змінна або щонайменше одна копія вибраної змінної також знаходиться у пристрої,
- а у фазі використання:
- в присутності пристрою, кожного разу, коли цього вимагає фрагмент першої виконуваної частини, використовують змінну або копію змінної, що знаходиться у пристрої так, щоб указаний фрагмент виконувався коректно й, отже, захищена програма була повнофункціональною,
- а за відсутності пристрою, незважаючи на запит фрагмента першої виконуваної частини на використання змінної або копії змінної, що знаходиться у пристрої, не забезпечується можливість коректної відповіді на вказаний запит, так що принаймні вказаний фрагмент не виконується коректно й, отже, захищена програма не є повнофункціональною. Відповідно до ще однієї кращої реалізації способу згідно з винаходом:
- у фазі захисту:
- визначають:
- як команду запуску - команду інструкції,
- як залежну функцію - інструкцію,
- як настановний параметр, - щонайменше один аргумент для команди запуску, що відповідає, щонайменше частково, інформації, переданій системою обробки даних на пристрій, щоб викликати запуск відповідної залежної функції,
- метод перейменування настановних параметрів, що дозволяє перейменувати настановні параметри, щоб одержати команди запуску з перейменованими настановними параметрами,
- і засоби відновлення, призначені для застосування у пристрої у фазі використання і що дозволяють знайти залежну функцію, що її необхідно виконати, виходячи з перейменованого настановного параметра;
- конструюють засоби експлуатації, що дозволяють пристрою задіяти засоби відновлення;
- і модифікують захищену програму:
- за допомогою вибору у вихідному коді захищеної програми команд запуску,
- за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми за допомогою перейменування настановних параметрів вибраних команд запуску, щоб приховати ідентичність відповідних залежних функцій,
- і за допомогою створення:
- першої частини об'єктного коду захищеної програми, причому перша частина об'єктного коду така, що в ході виконання захищеної програми виконуються команди запуску з перейменованими настановними параметрами,
- і другої частини об'єктного коду захищеної програми, що містить засоби експлуатації, що використовують також засоби відновлення, причому друга частина об'єктного коду така, що після завантаження в пристрій, у ході виконання захищеної програми ідентичність залежних функцій, виконання яких викликано першою виконуваною частиною, відновлюється за допомогою другої виконуваної частини, а залежні функції виконуються

UA 77187 C2
UA 77187 C2

за допомогою другої виконуваної частини,

- а у фазі використання:

- у присутності пристрою, кожного разу, коли цього вимагає команда запуску з перейменованими настановними параметрами, яка міститься у фрагменті першої виконуваної частини, відновлюють у пристрої ідентичність відповідної залежної функції і виконують її так, щоб указаний фрагмент виконувався коректно й, отже, захищена програма була повнофункціональною;

- тоді як за відсутності пристрою, незважаючи на запит фрагмента першої виконуваної частини на запуск виконання у пристрої залежної функції, не забезпечується можливість коректної відповіді на вказаний запит, так що принаймні вказаний фрагмент не виконується коректно й, отже, захищена програма не є повнофункціональною.

Відповідно до одного з варіантів здійснення способу згідно з винаходом:

- у фазі захисту:

- визначають, принаймні для однієї залежної функції, сімейство алгоритмічно еквівалентних залежних функцій, які викликаються командами запуску, перейменовані настановні параметри яких є різними;

- і модифікують захищену програму:

- за допомогою вибору у вихідному коді захищеної програми щонайменше однієї команди запуску з перейменованими настановними параметрами,

- і за допомогою модифікації принаймні одного вибраного фрагмента коду захищеної програми шляхом заміни щонайменше перейменованого настановного параметру команди запуску з вибраним настановним параметром на інший перейменований настановний параметр, що викликає запуск залежної функції з того ж сімейства.

Відповідно до варіанта реалізації спосіб згідно з винаходом включає:

- у фазі захисту визначення, щонайменше для однієї залежної функції сімейства, алгоритмічно еквівалентних залежних функцій,

- за допомогою зчеплення поля шумів з інформацією, що визначає ту функціональну частину залежної функції, яка виконується у пристрої,

- або за допомогою використання поля ідентифікації інструкції і полів ідентифікації, передбачених для операндів.

Відповідно до варіанта реалізації способу згідно з винаходом:

- у фазі захисту визначають:

- як метод перейменування настановних параметрів - метод кодування для кодування настановних параметрів,

- і як засоби відновлення - засоби, що застосовують метод декодування для розкодування перейменованих настановних параметрів і відновлення ідентичності залежних функцій, що їх слід виконати у пристрої.

Відповідно до іншої кращої форми реалізації способу згідно з винаходом:

- у фазі захисту:

- модифікують захищену програму:

- за допомогою вибору у вихідному коді захищеної програми щонайменше одного умовного переходу, виконуваного щонайменше в одному вибраному алгоритмі,

- за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми, причому зазначена модифікація така, що в ході виконання захищеної програми функціональна можливість щонайменше одного вибраного умовного переходу виконується, за допомогою другої виконуваної частини, у пристрої,

- і за допомогою створення:

- першої частини об'єктного коду захищеної програми, причому перша частина об'єктного коду така, що в ході виконання захищеної програми функціональна можливість щонайменше одного вибраного умовного переходу виконується у пристрої,

- і другої частини об'єктного коду захищеної програми, причому друга частина об'єктного коду така, що після завантаження у пристрій, у ході виконання захищеної програми реалізується друга виконувана частина, за допомогою якої виконується функціональна можливість принаймні одного вибраного умовного переходу,

- а у фазі використання:

- у присутності пристрою, кожного разу, коли цього вимагає фрагмент першої виконуваної частини, виконують у пристрої функції принаймні одного вибраного умовного переходу таким чином, щоб указаний фрагмент виконувався коректно й, отже, захищена програма була повнофункціональною;

- а за відсутності пристрою, незважаючи на запит фрагмента першої виконуваної частини на виконання функцій умовного переходу в пристрої, не забезпечується можливість коректної відповіді на вказаний запит, так що принаймні вказаний фрагмент не виконується коректно й, отже, захищена програма не є повнофункціональною.

Відповідно до наступного варіанту здійснення способу згідно з винаходом у фазі захисту модифікують захищену програму:

- за допомогою вибору у вихідному коді захищеної програми щонайменше однієї серії вибраних умовних переходів,

- за допомогою модифікації щонайменше одного вибраного фрагмента коду захищеної програми, причому згадана модифікація така, що в ході виконання захищеної програми глобальна функціональна можливість щонайменше однієї вибраної серії умовних переходів виконується у пристрої за допомогою другої виконуваної частини,

UA 77187 C2

- і за допомогою створення:

- першої частини об'єктного коду захищеної програми, причому перша частина об'єктного коду така, що в ході виконання захищеної програми функціональна можливість щонайменше однієї вибраної серії умовних переходів виконується у пристрої,

- і другої частини об'єктного коду захищеної програми, причому друга частина об'єктного коду така, що після завантаження у пристрій, у ході виконання захищеної програми реалізується друга виконувана частина, за допомогою якої виконується глобальна функціональна можливість принаймні однієї вибраної серії умовних переходів.

Спосіб згідно з винаходом дозволяє також захистити використання програми шляхом застосування пристрою обробки і зберігання, що його особливістю є здатність містити частину виконуваної програми. Звідси випливає, що будь-яка похідна версія програми, яка спробує функціонувати без пристрою обробки і зберігання, зажадає відтворення частини програми, що міститься у пристрої зберігання в ході виконання. Як наслідок ця похідна версія програми не буде повнофункціональною.

Інші різноманітні властивості винаходу стануть ясні з нижченаведеного опису, з посиланням на додані креслення, на яких показані як невичерпні приклади можливі варіанти і форми реалізації та використання винаходу.

Фіг.10 і 11 являють собою функціональні блок-схеми, що ілюструють різноманітні представлення програми, відповідно не захищеної і захищеної способом згідно з винаходом.

На Фіг.20-22 наведені як приклади різноманітні форми виконання пристрою для здійснення способу за винаходом.

Фіг.30 і 31 являють собою функціональні блок-схеми, що пояснюють загальний принцип способу згідно з винаходом.

Фіг.40-43 являють собою схеми, що ілюструють спосіб захисту згідно з винаходом, що реалізує принцип захисту за допомогою змінної.

Фіг.70-74 являють собою схеми, що ілюструють спосіб захисту згідно з винаходом, що реалізує принцип захисту за допомогою детектування й примусу.

Фіг.80-85 являють собою схеми, що ілюструють спосіб захисту згідно з винаходом, що реалізує принцип захисту за допомогою перейменування.

Фіг.90-92 являють собою схеми, що ілюструють спосіб захисту згідно з винаходом, що реалізує принцип захисту за допомогою умовного переходу.

Фіг.100 являє собою схему, що ілюструє різноманітні фази здійснення винаходу.

На Фіг.110 наведений приклад реалізації системи, що дозволяє реалізувати стадію побудови фази захисту згідно з винаходом.

На Фіг.120 наведений приклад реалізації пристрою передперсоналізації, використовуваного у способі захисту згідно з винаходом.

На Фіг.130 наведений приклад реалізації системи, що дозволяє здійснити стадію виготовлення засобів для фази захисту згідно з винаходом.

На Фіг.140 наведений приклад реалізації системи, що дозволяє застосувати спосіб захисту згідно з винаходом.

На Фіг.150 наведений приклад реалізації пристрою персоналізації, використовуваного у способі захисту згідно з винаходом.

Дані, що підтверджують можливість здійснення винаходу Надалі в описі використовуються такі визначення:

- Системою 3 обробки даних є система, здатна виконувати програму.
- Пристроєм обробки і зберігання є пристрій, здатний:
- одержувати дані, передані системою 3 обробки даних,
- повертати дані системі 3 обробки даних,
- зберігати дані в таємниці щонайменше частково, і зберігати щонайменше їхню частину навіть у випадку, коли пристрій відключений від живлення,
- і здійснювати алгоритмічну обробку даних, причому дана обробка частково або цілком є секретною.
- Пристроєм 6 є пристрій зберігання або пристрій обробки і зберігання, що реалізує спосіб згідно з винаходом.
- Незадіяним пристроєм 60 є пристрій, який не використовує спосіб згідно з винаходом, але який може одержати інформацію, що перетворює його на пристрій 6.
- Передперсоналізований пристрій 66 являє собою незадіяний пристрій 60, що одержав частину інформації, яка дозволяє йому, після одержання додаткової інформації, бути перетвореним на пристрій 6.
- Завантаження даних у незадіяний пристрій 60 або в передперсоналізований пристрій 66 відповідає передачі інформації в незадіяний пристрій 60 або в передперсоналізований пристрій 66 і зберіганню згаданих переданих даних. Передача інформації може містити в собі зміну її формату.
- Змінна, величина або функція, що міститься в системі 3 обробки даних, позначаються надалі заголовними буквами, а змінна, величина або функція, що міститься у пристрої 6, позначаються надалі малими літерами.
- "Захищеною програмою" є програма, що була захищена щонайменше на основі одного принципу захисту, реалізованого у способі згідно з винаходом.
- "Уразливою програмою" є програма, що не була захищена жодним з принципів захисту, реалізованих у способі згідно з винаходом.
- У випадку, коли розходження між уразливою і захищеною програмою несуттєве, застосовується термін "програма".

UA 77187 C2

- Програма може бути представлена в різноманітній формі відповідно до моменту її життєвого циклу, тобто як:

- вихідний код,

- об'єктний код,

- дистрибутив,

- динамічне представлення.

- Представлення програми у вигляді вихідного коду розуміється як представлення, що дає після перетворення представлення у вигляді об'єктного коду. Представлення у вигляді вихідного коду може подаватися на різноманітних рівнях, від абстрактного концептуального рівня до рівня, безпосередньо виконуваного системою обробки даних або пристроєм обробки і зберігання.

- Об'єктне представлення програми (представлення на рівні об'єктного коду) відповідає рівню представлення, на якому програма, після перенесення в дистрибутив і наступного завантаження в систему обробки даних або пристрій обробки і зберігання, може бути виконана. Це може бути, наприклад, двійковий код, інтерпретований код тощо.

- Дистрибутивом є фізичний або віртуальний носій, що містить об'єктне представлення, причому цей дистрибутив має надаватися в розпорядження користувача, щоб дозволити йому використовувати програму.

- Динамічне представлення відповідає виконанню програми з дистрибутива.

- Фрагмент програми відповідає певній її частині й може, наприклад, відповідати одній або декільком інструкціям (послідовним або ні) і/або одному або декільком функціональним блокам (послідовним або ні), і/або одній або декільком функціям, і/або одній або декільком підпрограмам, і/або одному або декільком модулям. Фрагмент програми може відповідати і всій програмі цілком.

На Фіг.10 і 11 наведені різноманітні представлення відповідно вразливої програми 2N у загальному вигляді і програми 2p, захищеної згідно з винаходом.

На Фіг.10 наведені різноманітні представлення вразливої програми 2v, що з'являються в ході її життєвого циклу. Вразлива програма 2v може з'являтися в одному з різноманітних виглядів, тобто як:

- вихідний код 2vs;

- об'єктний код 2vo;

- дистрибутив 2vd, який може надаватися звичайно на фізичному носії, наприклад на компакт-диску, або у вигляді файлів, переданих по мережі (за стандартом GSM, по мережі Інтернет тощо);

- у вигляді динамічного представлення 2ve, що відповідає виконанню вразливої програми 2v у системі 3 обробки даних будь-яких відомих типів, які в класичному випадку містять принаймні один процесор 4.

Фіг.11 ілюструє різноманітні представлення захищеної програми 2p, що з'являються в ході її життєвого циклу. Захищена програма 2p може також з'являтися у вигляді:

- вихідного коду (представлення) 2ps, що містить першу частину вихідного коду, призначену для системи 3 обробки даних, і, можливо, другу частину вихідного коду, призначену для пристрою 6, причому частина цих частин вихідного коду може звичайно міститися в загальних файлах;

- об'єктного представлення 2po, що містить першу частину 2pos об'єктного коду, призначену для системи 3 обробки даних, і, можливо, другу частину 2pou об'єктного коду, призначену для пристрою 6;

- дистрибутив у 2pd, що містить:

- першу частину 2pds дистрибутива, що містить першу частину 2pos об'єктного коду, причому ця перша частина 2pds дистрибутива призначена для системи 3 обробки даних і може бути представлена звичайно у формі дистрибутива на фізичному носії, наприклад на компакт-диску, або у вигляді файлів, переданих по мережі (за стандартом GSM, по мережі Інтернет тощо), - і другу частину 2pdu дистрибутива, поданого у вигляді:

- щонайменше одного незадіяного пристрою 60,

- або щонайменше одного передперсоналізованого пристрою 66, у який була завантажена частина другої частини 2pou об'єктного коду і для якого користувач мусить завершити персоналізацію шляхом завантаження додаткової інформації, щоб одержати пристрій 6, причому ця додаткова інформація може надходити, наприклад, шляхом завантаження або передачі по мережі,

- або щонайменше одного пристрою 6, у який була завантажена друга частина 2pou об'єктного коду;

- або у вигляді динамічного представлення 2pe, що відповідає виконанню захищеної програми 2p. Це динамічне представлення 2pe містить першу виконувану частину 2pes, що виконується в системі 3 обробки даних, і другу виконувану частину 2peu, що виконується у пристрої 6.

У випадку, коли розходження між різноманітними представленнями захищеної програми 2p несуттєве, використовуються вирази "перша частина захищеної програми" і "друга частина захищеної програми".

Реалізація способу згідно з винаходом відповідно до динамічного представлення, проілюстрованого на Фіг.11, використовує пристрій 1p захисту, що містить систему 3 обробки даних, сполучену лінією 5 зв'язку з пристроєм 6. Система 3 обробки даних може бути будь-якого типу і містить у звичайному варіанті щонайменше один процесор 4. Система 3 обробки даних може бути комп'ютером або бути частиною, наприклад, різноманітних машин, пристроїв, стаціонарних або рухливих виробів, у тому числі будь-яких транспортних засобів. Лінія 5 зв'язку може бути здійснена будь-яким можливим способом, наприклад, по лінії послідовної передачі, по шині USB, по радіо, по оптичному каналу, по мережі або через пряме електричне з'єднання зі схемою системи 3 обробки даних тощо. Слід зазначити, що пристрій 6 може фізично знаходитися всередині тієї ж інтегральної схеми, що й процесор 4 системи 3 обробки даних. У цьому випадку пристрій 6 може розглядатися як співпроцесор стосовно до процесора 4 системи обробки даних, а лінія 5 зв'язку є внутрішньою лінією зв'язку в інтегральній схемі.

UA 77187 C2

UA 77187 C2

На Фіг.20-22 наведені як приклади, що не вичерпують можливі варіанти, різноманітні форми реалізації пристрою 1p захисту, який дозволяє реалізувати спосіб захисту згідно з винаходом.

У прикладі реалізації за Фіг.20 пристрій 1p захисту містить, як систему 3 обробки даних, комп'ютер і, як пристрій 6, карту 7 з мікрочипом та її інтерфейс 8, звичайно називаний пристроєм читання карт. Комп'ютер 3 зв'язаний з пристроєм 6 за допомогою лінії 5 зв'язку. В ході виконання захищеної програми 2p перша виконувана частина 2pes, яка виконується в системі 3 обробки даних, і друга виконувана частина 2peu, яка виконується в карті 7 з мікрочипом і в її інтерфейсі 8, мають бути функціональними, щоб захищена програма 2p була повнофункціональною.

У прикладі реалізації за Фіг.21 пристрій 1p захисту міститься у виробі 9 загального виду, що містить різноманітні органи 10, адаптовані до функції або до функцій, реалізованих таким виробом 9. Пристрій 1p захисту містить, з одного боку, систему 3 обробки даних, умонтовану у виріб 9, а з другого боку - пристрій 6, зв'язаний з виробом 9. Щоб виріб 9 був повнофункціональним, захищена програма 2p має бути повністю функціональною. Так, у ході виконання захищеної програми 2p і перша виконувана частина 2pes, яка виконується в системі 3 обробки даних, і друга виконувана частина 2peu, яка виконується у пристрої 6, мають бути працездатні. Ця захищена програма 2p дозволяє, отже, непрямим чином захистити від неавторизованого використання виріб 9 або одну з його функціональних можливостей. Виріб 9 може бути, наприклад, установкою, системою, машиною, іграшкою, електропобутовим приладом, телефоном.

У прикладі реалізації за Фіг.22 пристрій 1p захисту містить множину комп'ютерів, а також частину комунікаційної мережі. Система 3 обробки даних являє собою перший комп'ютер, зв'язаний за допомогою лінії 5 зв'язку мережного типу з пристроєм 6, який являє собою другий комп'ютер. Для реалізації винаходу другий комп'ютер 6 використовується як сервер ліцензій для захищеної програми 2p. У ході виконання захищеної програми 2p і перша виконувана частина 2pes, яка виконується в першому комп'ютері 3, і друга виконувана частина 2peu, яка виконується в другому комп'ютері 6, мають бути функціональними, щоб захищена програма 2p була повнофункціональною.

Фіг.30 дозволяє пояснити точніше спосіб захисту згідно з винаходом. Слід зазначити, що вразлива програма 2v розглядається як виконувана повністю в системі 3 обробки даних. Навпаки, у випадку реалізації захищеної програми 2p система 3 обробки даних містить засоби 12 передачі, зв'язані лінією 5 зв'язку із засобами 13 передачі, що становлять частину пристрою 6, що дозволяє сполучатися між собою першій виконуваній частини 2pes і другій виконуваній частині 2peu захищеної програми 2p.

Слід мати на увазі, що засоби 12, 13 передачі реалізовані програмно або матеріально і здатні забезпечити й, можливо, оптимізувати передачу даних між системою 3 обробки даних і пристроєм 6. Ці засоби 12, 13 передачі пристосовані для того, щоб дозволити скористатися захищеною програмою 2p, яка є переважно незалежною від типу застосовуваної лінії 5 зв'язку. Ці засоби 12, 13 передачі не стосуються предмета винаходу і не описуються докладніше, тому що вони добре відомі фахівцям. Перша частина захищеної програми 2p містить команди. В ході виконання захищеної програми 2p виконання цих команд першою виконуваною частиною 2pes дозволяє здійснити зв'язок між першою виконуваною частиною 2pes і другою виконуваною частиною 2peu. Надалі в описі ці команди подані у вигляді IN, OUT або TRIG.

Як показано на Фіг.31, щоб дозволити реалізацію другої виконуваної частини 2peu захищеної програми 2p, пристрій 6 містить засоби 14 захисту. В разі, якщо пристрій 6 є запам'ятовуючим пристроєм, засоби 14 захисту містять засоби 15 запам'ятовування. У випадку, якщо пристрій 6 є пристроєм обробки і зберігання, засоби 14 захисту містять засоби 15 запам'ятовування і засоби 16 обробки.

Для спрощення подальшого опису будемо вважати, що в ході виконання захищеної програми 2p пристрій 6 є наявним або пристрій 6 відсутній. У дійсності, пристрій 6 у тому випадку, коли містить засоби 14 захисту, не пристосовані до виконання другої виконуваної частини 2peu захищеної програми 2p, також розглядається як відсутній щоразу, коли виконання захищеної програми 2p не є коректним. Іншими словами:

- пристрій 6, що фізично присутній і містить засоби 14 захисту, пристосовані до виконання другої виконуваної частини 2peu захищеної програми 2p, завжди розглядається як присутній;

- пристрій 6, що фізично присутній, але що містить непридатні засоби 14 захисту, тобто такі, що не дозволяють здійснити коректну реалізацію другої виконуваної частини 2peu захищеної програми 2p, розглядається як присутній, якщо він функціонує коректно, і як відсутній, якщо він не функціонує коректно;

- пристрій 6, фізично відсутній, завжди розглядається як відсутній.

У випадку, якщо пристрій 6 складається з карти 7 з мікрочипом і її інтерфейсу 8, засоби 13 передачі розділяються на дві частини, одна з яких знаходиться на інтерфейсі 8, а інша - на карті 7 з мікрочипом. У цьому прикладі реалізації відсутність карти 7 з мікрочипом розглядається як еквівалент відсутності пристрою 6. Іншими словами, за відсутності карти 7 з мікрочипом і/або її інтерфейсу 8 засоби 14 захисту недоступні й, отже, не дозволяють здійснити виконання другої виконуваної частини 2peu захищеної програми, так що захищена програма 2p не є повнофункціональною.

Згідно з винаходом, спосіб захисту спрямований на реалізацію принципу захисту, названого "детектування й примус", опис якого виконано з посиланням на Фіг.70-74.

Для реалізації принципу захисту за допомогою детектування й примусу визначаються:

- щонайменше одна характеристика виконання програми, яка може бути проконтрольована, принаймні, частково у пристрої 6;

- щонайменше один критерій, що його слід дотримуватися принаймні для однієї характеристики виконання програми;

- засоби 17 детектування, що їх необхідно застосовувати у пристрої 6 і які дозволяють виявити, що

UA 77187 C2

принаймні одна характеристика виконання програми не відповідає щонайменше одному відповідному критерію;

- засоби 18 примусу, що їх необхідно застосовувати у пристрої 6 і які дозволяють проінформувати систему 3 обробки даних і/або модифікувати виконання програми, поки не дотримано хоча б одного критерію.

Для реалізації принципу захисту за допомогою детектування й примусу конструюють також засоби експлуатації, які дозволяють перетворити незадіяний пристрій 60 на пристрій 6, що принаймні реалізує засоби 17 детектування й засоби 18 примусу.

На Фіг.70 показані засоби, необхідні для реалізації принципу захисту за допомогою детектування й примусу. Пристрій 6 містить засоби 17 детектування й засоби 18 примусу, що належать засобам 16 обробки. Засоби 18 примусу одержують інформацію про недотримання критерію від засобів 17 детектування.

Точніше кажучи, засоби 17 детектування використовують інформацію, що виходить від засобів 13 передачі і/або від засобів 15 запам'ятовування і засобів 16 обробки, щоб дотримувалася одна або декілька характеристик виконання програми. Кожній характеристиці виконання програми зіставляється принаймні один критерій, що його слід дотримуватися.

У випадку, якщо виявлено, що принаймні одна характеристика виконання програми не задовольняє щонайменше одному критерію, засоби 17 детектування інформують про це засоби 18 примусу. Ці засоби 18 примусу адаптовані для зміни відповідним чином стану пристрою 6.

Для реалізації принципу захисту за допомогою детектування й примусу мають бути вибрані також:

- щонайменше одна характеристика виконання контрольованої програми, серед характеристик виконання, що можуть бути проконтрольовані;

- щонайменше один критерій, що його слід дотримуватися принаймні для однієї вибраної характеристики виконання програми;

- у вихідному коді 2vs уразливої програми, - щонайменше один алгоритм, для якого потрібно контролювати щонайменше одну характеристику виконання програми;

- у вихідному коді 2vs уразливої програми, - щонайменше один фрагмент, що містить принаймні один вибраний алгоритм.

За виконання цих умов щонайменше один вибраний фрагмент коду 2vs уразливої програми модифікується, щоб одержати вихідний код 2ps захищеної програми. Ця модифікація така, що саме в ході виконання захищеної програми 2p:

- щонайменше один фрагмент першої виконуваної частини 2pes, яка виконується в системі 3 обробки даних, враховує, що принаймні одна характеристика виконання вибраної програми має бути проконтрольована, щонайменше частково, у пристрої 6;

- друга виконувана частина 2peu, яка виконується у пристрої 6, контролює, щонайменше частково, одну характеристику виконання вибраної програми.

У ході виконання програми 2p, захищеної за допомогою принципу детектування й примусу, в присутності пристрою 6 має місце така ситуація:

- якщо всіх критеріїв, що відповідають усім контрольованим характеристикам виконання всіх модифікованих фрагментів захищеної програми 2p, дотримано, ці модифіковані фрагменти захищеної програми 2p функціонують належним чином й, отже, захищена програма 2p функціонує належним чином;

- якщо ж щонайменше одного з критеріїв, що відповідає характеристиці контрольованого виконання одного фрагмента захищеної програми 2p, не дотримано, система 3 обробки даних інформується про це і/або функціонування фрагмента захищеної програми 2p модифікується таким чином, щоб функціонування захищеної програми 2p було змінене.

Природно, що за відсутності пристрою 6 щонайменше один запит одного фрагмента першої виконуваної частини 2pes захищеної програми 2p на використання пристрою 6 не може бути коректно виконаний, так що принаймні ця частина не виконується коректно й, отже, захищена програма 2p не є повнофункціональною.

Для реалізації принципу захисту за допомогою детектування й примусу бажано використовувати два типи характеристики виконання програми.

Перший тип характеристики виконання програми відповідає змінній контролю виконання програми, а другий тип відповідає профілю використання програми. Ці обидва типи характеристик можуть використовуватися незалежно або в поєднанні.

Для реалізації принципу захисту за допомогою детектування й примусу, що використовує як характеристику виконання програми змінну контролю виконання програми, мають визначатися:

- у засобах 15 запам'ятовування - можливість запам'ятати щонайменше одну змінну контролю, що служить як кількісна характеристика використання щонайменше однієї функціональної можливості програми;

- у засобах 17 детектування - можливість спостерігати щонайменше одне порогове значення, зв'язане з кожною змінною контролю;

- засоби поновлення, що дозволяють поновити кожну змінну контролю залежно від використання функціональної можливості, з якою вона зв'язана.

Конструюють також засоби експлуатації, які задіюють, окрім засобів 17 детектування і засобів 18 примусу, також засоби поновлення.

Крім того, у вихідному коді 2vs уразливої програми вибираються:

- щонайменше одна функціональна можливість уразливої програми 2v, що її використання можна проконтролювати за допомогою змінної для кількісного контролю;

- щонайменше одна змінна для кількісного контролю, що служить як кількісна характеристика згаданої функціональної можливості;

UA 77187 C2

- щонайменше одне порогове значення, зв'язане зі змінною для кількісного контролю і відповідне межі використання вказаної функціональної можливості;

- і щонайменше один метод поновлення змінної для кількісного контролю відповідно до використання зазначеної функціональної можливості.

Вихідний код 2vs уразливої програми потім модифікується, щоб одержати вихідний код 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p друга виконувана частина 2peu:

- поновлює значення змінної для кількісного контролю відповідно до використання згаданої функціональної можливості;

- і бере до уваги щонайменше одне перевищення порогового значення.

Іншими словами, в ході виконання захищеної програми 2p значення змінної для кількісного контролю поновлюється відповідно до використання згаданої функціональної можливості, й у разі перевищення порогового значення засобу 17 детектування інформують про це засоби 18 примусу, які приймають рішення, пристосоване для того, щоб проінформувати систему 3 обробки даних і/або модифікувати обробку, здійснювану засобами 16 обробки. Це дозволяє модифікувати функціонування фрагмента захищеної програми 2p таким чином, щоб функціонування захищеної програми 2p було змінено.

Для реалізації першого кращого варіанта реалізації принципу захисту за допомогою детектування й примусу, який використовує як характеристику змінну для кількісного контролю, визначають:

- щонайменше для однієї змінної для кількісного контролю - декілька відповідних порогових значень;

- і різноманітні засоби примусу, що відповідають кожному з цих порогових значень.

У вихідному коді 2vs уразливої програми вибираються також:

- щонайменше одна змінна для кількісного контролю, що служить як кількісна характеристика використання щонайменше однієї функціональної можливості програми, з якою мають зв'язуватися декілька порогових значень, що відповідають різноманітним межам використання зазначеної функціональної можливості;

- і щонайменше два порогових значення, зв'язані зі змінною для кількісного контролю.

Вихідний код 2vs уразливої програми потім модифікується, щоб одержати вихідний код 2ps захищеної програми. Ця модифікація така, що в ході виконання захищеної програми 2p друга виконувана частина 2peu:

- поновлює значення змінної для кількісного контролю відповідно до використання згаданої функціональної можливості;

- і по-різному враховує перевищення різноманітних порогових значень.

Іншими словами, у звичайному випадку в ході виконання захищеної програми 2p при перевищенні першого порогового значення пристрій 6 інформує систему 3 обробки даних, даючи команду захищеній програмі 2p більше не використовувати цю функціональну можливість. Якщо ж захищена програма 2p продовжує використовувати цю функціональну можливість, то може бути перевищене друге порогове значення. В разі перевищення другого порогового значення засоби 18 примусу можуть зробити непрацездатною вибрану функціональну можливість і/або зробити непрацездатною захищену програму 2p.

Для реалізації другого кращого варіанта принципу захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю, визначають засоби перезавантаження, що дозволяють щонайменше одне додаткове використання принаймні однієї функціональної можливості програми, контрольованої за допомогою змінної для кількісного контролю.

Конструюють також засоби експлуатації, де застосовані, крім засобів 17 детектування, засобів 18 примусу й засобів відновлення, також і засоби перезавантаження.

Крім того, у вихідному коді 2vs уразливої програми вибирається щонайменше одна змінна для кількісного контролю, що служить для обмеження використання щонайменше однієї функціональної можливості програми, для якої існує можливість дозволу на щонайменше одне додаткове використання.

Вихідний код 2vs уразливої програми потім модифікується, щоб одержати вихідний код 2ps захищеної програми, причому ця модифікація така, що у фазі, названій фазою перезавантаження, щонайменше одне додаткове використання щонайменше однієї функціональної можливості, що відповідає одній вибраній змінній для кількісного контролю, може бути дозволено.

У фазі перезавантаження відбувається поновлення щонайменше однієї вибраної змінної для кількісного контролю і/або щонайменше одного зв'язаного порогового значення, щоб дозволити принаймні одне додаткове використання відповідної функціональної можливості. Іншими словами, у фазі перезавантаження забезпечується можливість дозволити додаткові використання щонайменше однієї функціональної можливості захищеної програми 2p.

Для реалізації принципу захисту за допомогою детектування й примусу, що використовує як характеристику профіль використання програми, як критерій, що його слід дотримуватися для цього профілю використання, має бути визначена щонайменше одна ознака виконання програми.

Крім того, у вихідному коді 2vs уразливої програми вибирають:

- щонайменше один профіль використання, що його слід контролювати;

- і щонайменше одну ознаку виконання, що її має дотримуватися щонайменше один профіль використання.

Вихідний код 2vs уразливої програми потім модифікується, щоб одержати вихідний код 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p друга виконувана частина 2peu дотримується всіх вибраних ознак виконання.

Іншими словами, пристрій 6 саме контролює той спосіб, що ним виконується друга виконувана частина 2peu, і може інформувати систему 3 обробки даних і/або модифікувати функціонування захищеної програми 2p у

UA 77187 C2

UA 77187 C2

випадку, якщо не дотримана хоча б одна ознака.

У ході виконання програми 2p, захищеної на основі даного принципу, в присутності пристрою 6 має місце така ситуація:

- якщо всіх ознак виконання всіх модифікованих фрагментів захищеної програми 2p дотримано, то ці модифіковані фрагменти захищеної програми 2p функціонують належним чином, і, отже, захищена програма 2p функціонує належним чином;

- якщо ж хоча б однієї ознаки виконання одного фрагмента захищеної програми 2p не дотримано, про це інформується система 3 обробки даних і/або функціонування фрагмента захищеної програми 2p модифікується таким чином, щоб функціонування захищеної програми 2p було змінено.

Можна передбачити контроль різноманітних ознак виконання, наприклад контроль наявності інструкцій, що містять генератор міток, або контроль зчеплення виконання щонайменше однієї частини інструкцій.

Для реалізації принципу захисту за допомогою детектування й примусу, де як ознака виконання, що її слід дотримуватися, використовується контроль зчеплення виконання щонайменше частини інструкцій, визначають:

- набір інструкцій, інструкції зі складу якого можуть бути виконані у пристрої 6;

- набір команд інструкцій для цього набору інструкцій, причому ці команди інструкцій можуть бути виконані в системі 3 обробки даних. Виконання кожної з цих команд інструкцій у системі 3 обробки даних викликає у пристрої 6 виконання відповідної інструкції;

- засоби 17 детектування, що дозволяють виявити, що зчеплення інструкцій не відповідає бажаному;

- засоби 18 примусу, що дозволяють проінформувати систему 3 обробки даних і/або модифікувати виконання програми, якщо зчеплення інструкцій не відповідає бажаному.

Конструюють також засоби експлуатації, що дозволяють пристрою 6 виконувати інструкції з набору інструкцій, причому виконання цих інструкцій викликається виконанням команд інструкцій у системі 3 обробки даних.

Крім того, у вихідному коді 2vs уразливої програми вибирається щонайменше один алгоритм, який має бути винесений у пристрій 6 і для якого слід контролювати зчеплення щонайменше частини інструкцій.

Вихідний код 2vs уразливої програми потім модифікується, щоб одержати вихідний код 2ps захищеної програми. Ця модифікація така, що в ході виконання захищеної програми 2p:

- друга виконувана частина 2peu виконує щонайменше функціональну можливість вибраного алгоритму;

- вибраний алгоритм розкладається на інструкції;

- задане зчеплення, що його мають дотримуватися принаймні деякі з інструкцій у ході їхнього виконання у пристрої 6;

- перша виконувана частина 2pes захищеної програми 2p виконує команди інструкцій, які запускають виконання інструкцій у пристрої 6.

У ході виконання програми 2p, захищеної на основі цього принципу, в присутності пристрою 6 має місце така ситуація:

- якщо зчеплення інструкцій усіх модифікованих фрагментів захищеної програми 2p відповідає бажаному, ці модифіковані фрагменти захищеної програми 2p функціонують належним чином і, отже, захищена програма 2p функціонує належним чином;

- якщо ж зчеплення інструкцій фрагмента захищеної програми 2p, виконуваних у пристрої 6, не відповідає бажаному, то система 3 обробки даних інформується про це і/або функціонування фрагмента захищеної програми 2p модифікується таким чином, щоб функціонування захищеної програми 2p було змінено.

На Фіг.71 наведений приклад реалізації принципу захисту за допомогою детектування й примусу, де як ознака виконання, що її слід дотримуватися, використовується контроль зчеплення виконання щонайменше частини інструкцій у випадку, якщо бажаного зчеплення дотримано.

Перша виконувана частина 2pes захищеної програми 2p, виконувана в системі 3 обробки даних, виконує команди Cl_i інструкцій, що викликає у пристрої 6 виконання інструкцій i_i, що належать до набору інструкцій. У наборі інструкцій щонайменше деякі з інструкцій містять частину, що задає функціональну можливість інструкції, і частину, що дозволяє перевіряти бажане зчеплення для виконання інструкцій. У цьому прикладі команди Cl_i інструкцій подані як $TRIG(i_i)$, а бажаним зчепленням для виконання інструкцій є i_n, i_{n+1} і i_{n+2}. Виконання у пристрої 6 інструкції i_n дає результат a, а виконання інструкції i_{n+1} дає результат b. Інструкція i_{n+2} використовує як операнд результати a і b інструкцій i_n і i_{n+1}, а її виконання дає результат c.

З урахуванням того, що це зчеплення інструкцій, виконуваних у пристрої 6, відповідає бажаному, функціонування захищеної програми 2p відповідає нормальному або номінальному режиму.

На Фіг.72 наведений приклад реалізації принципу захисту за допомогою детектування й примусу, де як ознака виконання, що її слід дотримуватися, використовується контроль зчеплення виконання щонайменше частини інструкцій у випадку, якщо бажаного зчеплення не дотримано.

Відповідно до цього приклада бажаним зчепленням для виконання інструкцій є завжди i_n, i_{n+1} і i_{n+2}. Проте зчеплення виконання інструкцій модифікується заміною інструкції i_n на інструкцію i'_n таким чином, що дійсно виконуваним зчепленням є i'_n, i_{n+1} і i_{n+2}. Виконання інструкції i'_n дає результат a, тобто той самий результат, що й виконання інструкції i_n. Проте не пізніше, ніж при виконанні інструкції i_{n+2}, засоби 17 детектування виявляють, що інструкція i'_n не відповідає бажаній інструкції для вироблення результату a, використовуваного як операнд для інструкції i_{n+2}. Засоби 17 детектування інформують про це засоби 18 примусу, що модифікують, як наслідок, функціонування інструкції i_{n+2}, таким чином, що виконання інструкції i_{n+2} дає результат c', який може відрізнятися від c.

Зрозуміло, якщо виконання інструкції i'_n дає результат a', відмінний від результату a інструкції i_n, ясно, що

UA 77187 C2

UA 77187 C2

результат виконання інструкції i_{n+2} може також відрізнятися від с.

Отже, якщо зчеплення виконання інструкцій, виконуваних у пристрої 6, не відповідає бажаному, можна одержати модифікацію функціонування захищеної програми 2p.

На Фіг.73 і 74 показаний кращий варіант реалізації принципу захисту за допомогою детектування й примусу, де як ознака виконання, що її слід дотримуватися, використовується контроль зчеплення виконання щонайменше частини інструкцій. Відповідно до цього кращого варіанта визначається набір інструкцій, у якому щонайменше деякі інструкції працюють на регістрах і використовують щонайменше один операнд для видачі результату.

Як показано на Фіг.73, щонайменше для частини інструкцій, що працюють на регістрах, визначена частина PF, що задає функціональну можливість інструкції, і частина PE, що задає бажане зчеплення для виконання інструкцій. Частина PF відповідає коду операцій, відомому фахівцеві. Частина PE, що визначає бажане зчеплення, містить відповідні бітові поля:

- полю CII ідентифікації інструкції;
- і для кожного операнда k інструкції, де k пробігає значення від 1 до K, K - кількість операндів інструкції:
- полю CD_k прапора, що вказує, чи варто перевіряти походження операнда k,
- і полю CIP_k ідентифікації, передбаченої для операнда й що вказує на очікувану ідентичність інструкції, яка згенерувала вміст операнда k.

Як показано на Фіг.74, набір інструкцій містить V регістрів, що належать засобам 16 обробки, де кожний регістр названий R_v (v пробігає значення від 1 до V). Для кожного регістра R_v визначаються два поля, як-от:

- функціональне поле CF_v, відоме фахівцеві і що дозволяє зберігати результат виконання інструкцій;
- і поле CIG_v генерованої ідентифікації, в якому автоматично запам'ятовується ідентифікація останньої інструкції, що повернула свій результат у вказаний регістр, тобто такої, що згенерувала вміст функціонального поля CF_v. Це поле CIG_v генерованої ідентифікації поновлюється автоматично разом із вмістом поля CII ідентифікації інструкції, згенерувавши функціональне поле CF_v. Поле CIG_v генерованої ідентифікації не є ні доступним, ані таким, що модифікується для будь-якої іншої інструкції, і служить винятково для засобів 17 детектування.

В ході виконання інструкції засоби 17 детектування виконують для кожного операнда k такі операції:

- зчитується поле CD_k прапора;
- якщо цього вимагає поле CD_k прапора, то зчитуються обидва поля: поле CIP_k передбаченої ідентифікації і поле CIG_v генерованої ідентифікації, що відповідає регістрам, використовуваним операндом k;
- перевіряється рівність обох полів CIP_k і CIG_v;
- і якщо рівність хибна, то засоби 17 детектування вважають, що зчеплення виконання інструкцій не дотримано.

Засоби 18 примусу мають дозволяти модифікувати результат виконання інструкцій, коли засоби 17 детектування проінформують їх про недотримання зчеплення інструкцій. Краща реалізація полягає в тому, щоб модифікувати функціональну частину PF виконуваної інструкції або функціональної частини PF наступних інструкцій.

Відповідно до іншої кращої характеристики винаходу, спосіб захисту спрямований на реалізацію так званого принципу захисту за допомогою змінної, опис якого подано з посиланням на Фіг.40-43.

Для реалізації принципу захисту за допомогою змінної у вихідному коді 2vs уразливої програми вибирається щонайменше одна змінна, яка в ході виконання вразливої програми 2v частково визначає її стан. Під станом програми розуміється сукупність інформації, на даний момент необхідної для повного виконання цієї програми. Таким чином, відсутність згаданої вибраної змінної перешкоджає повному виконанню цієї програми. Вибирається також щонайменше один фрагмент вихідного коду 2vs уразливої програми, що містить щонайменше одну вибрану змінну.

Щонайменше один вибраний фрагмент коду 2vs уразливої програми в цьому випадку модифікується, щоб одержати вихідний код 2ps захищеної програми. Ця модифікація така, що в ході виконання захищеної програми 2p щонайменше один фрагмент першої виконуваної частини 2pes, який виконується в системі 3 обробки даних, враховує, що принаймні одна вибрана змінна або щонайменше одна копія вибраної змінної знаходиться у пристрої 6.

На Фіг.40 наведений приклад представлення вразливої програми 2v. У цьому прикладі в ході виконання вразливої програми 2v у системі 3 обробки даних мають місце:

- у момент t_1 присвоєння значення X змінній V_1, що подано як $V_1 \leftarrow X$;
- у момент t_2 присвоєння значення змінної V_1 змінній Y, що подано як $Y \leftarrow Y_1$;
- у момент t_3 присвоєння значення змінної V_1 змінній Z, що подано як $Z \leftarrow V_1$.

На Фіг.41 наведений приклад першої форми реалізації винаходу, для якої змінна знаходиться у пристрої 6. У цьому прикладі в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p в присутності пристрою 6 здійснюються:

- у момент t_1 виконання команди передачі, що викликає передачу даного X від системи 3 обробки даних до змінної v_1, розташованої в засобах 15 запам'ятовування пристрою 6, причому ця команда передачі подана як $OUT(v_1, X)$ і відповідає по завершенні присвоєнню значення X змінній v_1;
- у момент t_2 виконання команди передачі, що викликає передачу значення змінної v_1, що знаходиться у пристрої 6, системі 3 обробки даних, щоб присвоїти її значення змінній Y, причому ця команда передачі подана як $IN(v_1)$ і відповідає по завершенні присвоєнню значення v_1 змінній Y;
- у момент t_3 виконання команди передачі, що викликає передачу значення змінної v_1, що знаходиться у

UA 77187 C2

пристрої 6, системі 3 обробки даних, щоб присвоїти її значення змінній Z, причому ця команда передачі подана як IN(v_1) і відповідає по завершенні присвоєнню значення v_1 змінній Z.

Слід зазначити, що в ході виконання захищеної програми 2p щонайменше одна змінна знаходиться у пристрої 6. Так, коли цього вимагає частина першої виконуваної частини 2pes захищеної програми 2p, в присутності пристрою 6, значення цієї змінної, що знаходиться у пристрої 6, передається системі 3 обробки даних, щоб бути використаною першою виконуваною частиною 2pes захищеної програми 2p таким чином, щоб ця частина виконувалася коректно й, отже, захищена програма 2p була повнофункціональною.

На Фіг.42 наведений приклад другої форми реалізації винаходу, для якої копія змінної знаходиться у пристрої 6. У цьому прикладі в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p в присутності пристрою 6 здійснюються:

- у момент t_1 присвоєння значення X змінній V_1, що знаходиться в системі 3 обробки даних, а також виконання команди передачі, що викликає передачу даного X від системи 3 обробки даних до змінної v_1, розташованої в засобах 15 запам'ятовування пристрою 6, причому ця команда передачі подана як OUT(v_1, X);
- у момент t_2 - присвоєння значення змінної V_1змінній Y;
- у момент t_3 - виконання команди передачі, що викликає передачу значення змінної v_1, що знаходиться у пристрої 6, системі 3 обробки даних, щоб присвоїти її значення змінній Z, причому ця команда передачі подана як IN(v_1).

Слід зазначити, що в ході виконання захищеної програми 2p щонайменше одна копія однієї змінної знаходиться у пристрої 6.

Так, коли цього вимагає частина першої виконуваної частини 2pes захищеної програми 2p, у присутності пристрою 6, значення цієї копії змінної, що знаходиться у пристрої 6, передається системі 3 обробки даних, щоб бути використаною першою виконуваною частиною 2pes захищеної програми 2p таким чином, щоб ця частина виконувалася коректно й, отже, захищена програма 2p була повнофункціональною.

На Фіг.43 наведений приклад спроби виконання захищеної програми 2p за відсутності пристрою 6. У цьому прикладі в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p:

- у момент t_1 виконання команди передачі OUT(v_1, X) не може викликати передачу даного X змінній v_1 з огляду на відсутність пристрою 6;
- у момент t_2 виконання команди передачі IN(v_1) не може викликати передачу значення змінної v_1 системі 3 обробки даних з огляду на відсутність пристрою 6;
- у момент t_3 виконання команди передачі IN(v_1) не може викликати передачу значення змінної v_1 системі 3 обробки даних з огляду на відсутність пристрою 6.

Таким чином, представляється, що за відсутності пристрою 6 щонайменше один запит одного фрагмента першої виконуваної частини 2pes на використання змінної або копії змінної, що знаходиться у пристрої 6, не може бути коректно виконаний, отже, щонайменше ця частина не виконується коректно, й, отже, захищена програма 2p не є повнофункціональною.

Слід відзначити, що процеси передачі даних між системою 3 обробки даних і пристроєм 6 використовують тільки прості присвоєння (що проілюстровано на вищенаведених прикладах). Проте фахівець зможе скомбінувати їх з іншими операціями, щоб одержати складні операції, наприклад, OUT(v_1, 2*X+3) або $Z \leftarrow (5*v_1+v2)$.

Відповідно до іншої переважної характеристики винаходу спосіб захисту націлений на реалізацію принципу захисту, названого "перейменування", опис якого виконано з посиланням на Фіг.80-85.

Для реалізації принципу захисту перейменуванням мають визначатися:

- ансамбль залежних функцій, залежні функції якого можуть бути виконані за допомогою другої виконуваної частини 2peu у пристрої 6, можливо, із наступною передачею даних між системою 3 обробки даних і пристроєм 6 (причому цей ансамбль залежних функцій може бути кінцевим чи ні);
- ансамбль команд запуску для цих залежних функцій, причому ці команди запуску можуть виконуватися в системі 3 обробки даних і викликати у пристрої 6 виконання відповідних залежних функцій;
- для кожної команди запуску - настановний параметр, що відповідає, щонайменше частково, інформації, переданій першою виконуваною частиною 2pes другій виконуваній частині 2peu, щоб викликати запуск відповідної залежної функції, причому цей настановний параметр представляється у формі принаймні аргументу команди запуску;
- метод перейменування настановних параметрів, призначений для застосування в ході модифікації вразливої програми і дозволяє перейменувати настановні параметри таким чином, щоб одержати команди запуску з перейменованими параметрами, дозволяючи приховати ідентичність відповідних залежних функцій;
- засоби 20 відновлення, призначені для застосування у пристрої 6 у фазі використання і що дозволяють відновити початкові настановні параметри виходячи з перейменованих настановних параметрів, щоб знайти залежну функцію, що її необхідно виконати.

Для реалізації принципу захисту перейменуванням конструюють також засоби експлуатації, що дозволяють перетворити незадіяний пристрій 60, що містить засоби 15 запам'ятовування і засоби 16 обробки, на пристрій 6, що застосовує щонайменше засоби 20 відновлення.

Для реалізації принципу захисту перейменуванням у вихідному коді 2vs уразливої програми мають також вибиратися:

- щонайменше один алгоритм, що використовує щонайменше один операнд і що видає принаймні один результат;
- і щонайменше один фрагмент вихідного коду 2vs уразливої програми, що містить принаймні один вибраний

UA 77187 C2

алгоритм.

Вихідний код 2vs уразливої програми потім модифікується, щоб одержати вихідний код 2ps захищеної програми. Ця модифікація така, що:

- у ході виконання захищеної програми 2p щонайменше один фрагмент першої виконуваної частини 2pes, яка виконується в системі 3 обробки даних, враховує, що функціональна можливість щонайменше одного вибраного алгоритму виконується у пристрої 6;
- у ході виконання захищеної програми 2p друга виконувана частина 2peu, яка виконується у пристрої 6, виконує принаймні функціональну можливість щонайменше одного вибраного алгоритму;
- кожний вибраний алгоритм розкладається таким чином, що в ході виконання захищеної програми 2p кожний вибраний алгоритм виконується за допомогою другої виконуваної частини 2peu, використовуючи залежні функції. Кожний вибраний алгоритм переважно розкладається на залежні функції fd_n (де n пробігає значення від 1 до N), як-от:
- можливо, на одну або декілька залежних функцій, що дозволяють надати один або декілька операндів для пристрою 6,
- на залежні функції, деякі з яких використовують один або декілька операндів і які в поєднанні здійснюють функціональну можливість вибраного алгоритму, що використовує ці операнди,
- і, можливо, на одну або декілька залежних функцій, що дозволяють за допомогою пристрою 6 надати системі 3 обробки даних результат виконання вибраного алгоритму;
- у ході виконання захищеної програми 2p друга виконувана частина 2peu виконує залежні функції fd_n;
- у ході виконання захищеної програми 2p ці залежні функції запускаються командами запуску з перейменованими настановними параметрами;
- і впорядкування команд запуску вибране серед ансамблю впорядкувань, що дозволяють виконання захищеної програми 2p.

Перша виконувана частина 2pes захищеної програми 2p, виконувана в системі 3 обробки даних, виконує команди запуску з перейменованими настановними параметрами, що передають у пристрій 6 перейменовані настановні параметри. Це викликає у пристрої 6 відновлення за допомогою засобів 20 відновлення настановних параметрів, а потім виконання, за допомогою другої виконуваної частини 2peu, кожної з визначених вище залежних функцій fd_n.

Іншими словами, принцип захисту за допомогою перейменування полягає в тому, щоб перейменувати настановні параметри команд запуску, щоб одержати команди запуску з перейменованими настановними параметрами, виконання яких у системі 3 обробки даних викликає у пристрої 6 виконання залежних функцій, що запускалися б командами запуску з неперейменованими настановними параметрами, проте без того, щоб вивчення захищеної програми 2p дозволило визначити ідентичність виконуваних залежних функцій.

На Фіг.80 наведений приклад виконання вразливої програми 2v. У цьому прикладі в ході виконання вразливої програми 2v у системі 3 обробки даних у даний момент часу має місце розрахунок $Z \leftarrow F(X, Y)$, що відповідає присвоєнню змінній Z результату виконання алгоритму, поданого функцією F і який використовує операнди X і Y.

На Фіг.81 і 82 наведений приклад реалізації винаходу. На Фіг.81 наведений приклад часткової реалізації винаходу. Відповідно до цього приклада в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p у присутності пристрою 6 виконуються:

- у моменти t_1, t_2 - виконання елементарних команд CD_1, CD_2, що викликає у пристрої 6 виконання, за допомогою другої виконуваної частини 2peu, відповідних залежних функцій fd_1, fd_2, що забезпечують передачу даних X, Y з системи 3 обробки даних в області пам'яті відповідно x і y, розташовані в засобах 15 запам'ятовування пристрою 6, причому ці команди CD_1, CD_2 запуску подані відповідно як OUT(x, X), OUT (y, Y);
- у моменти з t_3 до t_{N-1} - виконання команд запуску з CD_3 до CD_{N-1}, що викликає у пристрої 6 виконання за допомогою другої виконуваної частини 2peu відповідних залежних функцій з fd_3 до fd_{N-1}, причому ці команди запуску з CD_3 до CD_{N-1} подані відповідно як TRIG(fd_3)-TRIG(fd_{N-1}), причому послідовність залежних функцій з fd_3 до fd_{N-1}, виконуваних у поєднанні, алгоритмічно еквівалентна функції F (точніше кажучи, виконання цих команд запуску приводить до виконання у пристрої 6 залежних функцій з fd_3 до fd_{N-1}, які використовують уміст областей пам'яті x, y і повертають результат в область пам'яті z пристрою 6);
- а в момент t_N здійснюється виконання команди CD_N запуску, що викликає у пристрої 6 виконання, за допомогою другої виконуваної частини 2peu, залежної функції fd_N, яка забезпечує передачу результату виконання алгоритму, що міститься в області пам'яті z пристрою 6, системі 3 обробки даних, щоб присвоїти його змінній Z. Ця команда подана як IN(z).

У даному прикладі, щоб повністю реалізувати винахід, як настановний параметр вибраний перший аргумент команд OUT запуску й аргумент команд TRIG і IN запуску. Вибрані в такий спосіб настановні параметри перейменовуються за методом перейменування настановних параметрів. Таким чином, настановні параметри команд запуску з CD_1 до CD_N, як-от: x, y, fd_3, fd_{N-1}, z перейменовуються так, щоб одержати відповідно R(x), R(y), $R(fd_3)$..., $R(fd_{N-1})$, R(z).

На Фіг.82 наведений приклад повної реалізації винаходу. Відповідно до цього приклада в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p, у присутності пристрою 6, здійснюються:

- у моменти t_1, t_2 - виконання команд $CDCR_1$, $CDCR_2$ запуску з перейменованими настановними параметрами, що передають у пристрій 6 перейменовані настановні параметри R(x), R(y), а також дані X, Y, що викликає у пристрої 6 відновлення (за допомогою засобів 20 відновлення) перейменованих настановних параметрів, щоб відновити настановні параметри, як-от ідентичність областей пам'яті x, y, а потім виконання, за допомогою

UA 77187 C2

другої виконуваної частини 2peu, відповідних залежних функцій fd_1, fd_2, які забезпечують передачу даних X, Y від системи 3 обробки даних до областей пам'яті відповідно x, y, розташованих у засобах 15 запам'ятовування пристрою 6 (ці команди $CDCR_1$, $CDCR_2$ запуску з перейменованими настановними параметрами представлені відповідно як OUT (R(x), X), OUT (R(y), Y));

- у моменти з t_3 до t_{N-1} - виконання команд $CDCR_3$ до $CDCR_{N-1}$ запуску з перейменованими настановними параметрами, що передають у пристрій 6 перейменовані настановні параметри з $R(fd_3)$ до $R(fd_{N-1})$, що викликає у пристрої 6 відновлення за допомогою засобів 20 відновлення настановних параметрів, як-от: з fd_3 до fd_{N-1}, а потім виконання, за допомогою другої виконуваної частини 2peu, залежних функцій з fd_3 до fd_{N-1}, причому ці команди (з $CDCR_3$ до $CDCR_{N-1}$) запуску з перейменованими настановними параметрами подані відповідно командами TRIG ($R(fd_3)$) до TRIG ($R(fd_{N-1})$);

- у момент t_N - виконання команди $CDCR_N$ запуску з перейменованими настановними параметрами, що передає у пристрій 6 перейменовані настановні параметри R(z). Це викликає у пристрої 6 відновлення, за допомогою засобів 20 відновлення, настановних параметрів, а саме ідентичності області пам'яті z, а потім виконання, за допомогою другої виконуваної частини 2peu, залежної функції fd_N, що забезпечує передачу результату виконання алгоритму, що міститься в області пам'яті z пристрою 6, системі 3 обробки даних, щоб присвоїти його змінній Z. Ця команда $CDCR_N$ запуску з перейменованими настановними параметрами подана як IN(R(z)).

У наведеному прикладі команди запуску з перейменованими настановними параметрами з 1 до N виконуються послідовно. Слід зазначити, що можна зробити два такі вдосконалення.

- Перше вдосконалення стосується випадку, коли декілька алгоритмів винесені у пристрій 6 і щонайменше результат виконання одного алгоритму використовується іншим алгоритмом (у цьому випадку деякі команди запуску з перейменованими настановними параметрами, що служать для передачі, можуть бути виключені).

- Друге вдосконалення має на меті належне впорядкування команд запуску з перейменованими настановними параметрами серед ансамблю впорядкувань, що дозволяють виконання захищеної програми 2p.

У цьому відношенні бажано вибрати таке впорядкування команд запуску з перейменованими настановними параметрами, що розділяє в часі виконання залежних функцій, уставляючи між ними ділянки коду, виконуваного в системі 3 обробки даних, і який при цьому містить (або не містить) команди запуску з перейменованими настановними параметрами, що служать для визначення інших даних. Фіг.83 і 84 ілюструють принцип такої реалізації.

На Фіг.83 наведений приклад виконання вразливої програми 2v. У цьому прикладі в системі 3 обробки даних у ході виконання вразливої програми 2v відбувається виконання двох алгоритмів, що приводять до визначення Z і Z' таких, що $Z \leftarrow F(X, Y)$ і $Z' \leftarrow F(X', Y')$.

На Фіг.84 наведений приклад реалізації способу згідно з винаходом, у якому обидва вибраних на Фіг.83 алгоритми винесені у пристрій 6. Відповідно до цього приклада в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p, у присутності пристрою 6, має місце, як пояснюється вище, виконання команд (з $CDCR_1$ до $CDCR_N$) запуску з перейменованими настановними параметрами, що відповідає визначенню Z, і виконання команд (з $CDCR_1'$ до $CDCR_M'$) запуску з перейменованими настановними параметрами, що відповідає визначенню Z'. Як показано, команди запуску з $CDCR_1$ до $CDCR_N$ не виконуються послідовно, оскільки з ними чергуються команди запуску з $CDCR_1'$ до $CDCR_M'$, а також інші фрагменти коду. В цьому прикладі, таким чином, реалізоване таке впорядкування: $CDCR_1$, уставлений фрагмент коду, $CDCR_1'$, $CDCR_2$, уставлений фрагмент коду, $CDCR_2'$, $CDCR_3'$, уставлений фрагмент коду, $CDCR_4'$, $CDCR_3$, $CDCR_4$,..., $CDCR_N$, $CDCR_M'$.

Слід зазначити, що в ході виконання фрагмента першої виконуваної частини 2pes захищеної програми 2p команди запуску з перейменованими настановними параметрами, що виконуються в системі 3 обробки даних, викликають у пристрої 6 відновлення ідентичності відповідних залежних функцій, а потім їхнє виконання. Таким чином, у присутності пристрою 6 цей фрагмент виконується коректно й, отже, захищена програма 2p є повнофункціональною.

На Фіг.85 наведений приклад спроби виконання захищеної програми 2p за відсутності пристрою 6. У цьому прикладі в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p виконання команди запуску з перейменованими настановними параметрами жодного моменту не може викликати ні відновлення настановних параметрів, ані виконання відповідної залежної функції через відсутність пристрою 6. Значення, що його необхідно присвоїти змінній Z, отже, не може бути визначене коректно.

Таким чином, вважається, що за відсутності пристрою 6 щонайменше один запит одного фрагмента першої виконуваної частини 2pes захищеної програми 2p на запуск відновлення настановних параметрів і виконання у пристрої залежної функції у пристрої 6 не може бути коректно виконаний, так що принаймні ця частина не виконується коректно й, отже, захищена програма 2p не є повнофункціональною.

Завдяки цьому принципу захисту за допомогою перейменування вивчення в захищеній програмі 2p команд запуску з перейменованими настановними параметрами не дозволяє визначити ідентичність залежних функцій, які мають виконуватися у пристрої 6. Слід зазначити, що перейменування настановних параметрів здійснюється в ході перетворення вразливої програми 2v на захищену програму 2p.

Відповідно до варіанта принципу захисту за допомогою перейменування має бути визначено, щонайменше для однієї залежної функції, сімейство алгоритмічно еквівалентних залежних функцій, що викликаються командами запуску з різноманітними перейменованими настановними параметрами. В цьому варіанті щонайменше один алгоритм, що використовує залежні функції, розкладається на залежні функції так, що принаймні одна з них заміняється залежною функцією того ж сімейства, замість того, щоб зберігати декілька

входжень тієї самої залежної функції. З цією метою команди запуску з перейменованими настановними параметрами модифікуються, щоб урахувати заміну залежних функцій на залежні функції того ж сімейства.

Іншими словами, дві залежні функції з одного сімейства мають різні настановні параметри й, отже, команди запуску з різними перейменованими настановними параметрами. Тому при вивченні захищеної програми 2p не є можливим виявити, чи є викликані залежні функції алгоритмічно еквівалентними.

Відповідно до першої кращої реалізації варіанта принципу захисту за допомогою перейменування визначається, щонайменше для однієї залежної функції, алгоритмічно еквівалентне сімейство залежних функцій, за допомогою зчеплення поля шумів з інформацією, що визначає ту функціональну частину залежної функції, що виконується у пристрої 6.

Відповідно до другої кращої реалізації варіанта принципу захисту за допомогою перейменування щонайменше для однієї залежної функції визначається алгоритмічно еквівалентне сімейство залежних функцій з використанням полів ідентифікації.

Відповідно до кращого варіанта реалізації принципу захисту за допомогою перейменування як методу перейменування настановних параметрів задається метод кодування, що дозволяє кодувати настановні параметри для їхнього перетворення на перейменовані настановні параметри. Слід нагадати, що перейменування настановних параметрів здійснюється у фазі P захисту. Для цього кращого варіанта засоби 20 відновлення являють собою засоби, що застосовують метод декодування, що дозволяє декодувати перейменовані настановні параметри і відновити таким чином ідентичність залежних функцій, що їх слід виконати у пристрої 6. Ці засоби відновлення застосовуються у пристрої 6 і можуть бути як програмними, так і апаратними. Засоби 20 відновлення запитуються у фазі U використання щоразу, коли команда запуску з перейменованими настановними параметрами виконується в системі 3 обробки даних з метою викликати у пристрої 6 виконання залежної функції.

Відповідно до іншої кращої характеристики винаходу спосіб захисту спрямований на реалізацію принципу захисту, названого "умовним переходом", опис якого проілюстровано Фіг.90-92.

Для реалізації принципу захисту за допомогою умовного переходу у вихідному коді 2vs уразливої програми вибирається щонайменше один умовний перехід BC. Вибирається також щонайменше один фрагмент вихідного коду 2vs уразливої програми, яка містить щонайменше один вибраний умовний перехід BC.

У даному варіанті щонайменше один вибраний фрагмент коду 2vs уразливої програми модифікується, щоб одержати вихідний код 2ps захищеної програми. Ця модифікація така, що в ході виконання захищеної програми 2p:

- щонайменше один фрагмент першої виконуваної частини 2pes, яка виконується в системі 3 обробки даних, ураховує те, що функціональна можливість щонайменше одного вибраного умовного переходу BC виконується у пристрої 6;

- друга виконувана частина 2peu, яка виконується у пристрої 6, виконує щонайменше функціональну можливість щонайменше одного вибраного умовного переходу BC і надає системі 3 обробки даних інформацію, що дозволяє першій виконуваній частині 2pes продовжити своє виконання у вибраному місці.

Перша виконувана частина 2pes захищеної програми 2p, виконувана в системі 3 обробки даних, виконує команди умовних переходів, що викликає у пристрої 6 виконання, за допомогою другої виконуваної частини 2peu, винесених умовних переходів bc, функціональні можливості яких еквівалентні функціональним можливостям вибраних умовних переходів BC. Для реалізації принципу захисту за допомогою умовного переходу пристрій 6 містить засоби 15 запам'ятовування і засоби 16 обробки.

На Фіг.90 наведений приклад виконання вразливої програми 2v. У цьому прикладі в ході виконання вразливої програми 2v у системі 3 обробки даних у даний момент часу має місце умовний перехід BC, що вказує вразливій програмі 2v місце, де варто продовжити її виконання, а саме одне з трьох можливих місць B_1, B_2 або B_3. Слід розуміти, що умовний перехід BC приймає рішення продовжити виконання програми в місці B_1, B_2 або B_3.

На Фіг.91 наведений приклад реалізації винаходу, в якому умовний перехід, відібраний для переносу в пристрій 6, відповідає умовному переходові BC.

У цьому прикладі в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p у присутності пристрою 6 мають місце:

- у момент t_1 - виконання команди CBC_1 умовного переходу, що викликає у пристрої 6 виконання, за допомогою другої виконуваної частини 2peu, винесеного умовного переходу bc, алгоритмічно еквівалентного умовному переходові BC, причому ця команда CBC_1 умовного переходу подана як TRIG(bc);

- і у момент t_2 - передача пристроєм 6 системі 3 обробки даних інформації, що дозволяє першій виконуваній частини 2pes продовжити своє виконання у вибраному місці, а саме B_1, B_2 або B_3.

Слід зазначити, що в ході виконання фрагмента першої виконуваної частини 2pes захищеної програми 2p команди умовних переходів, які виконуються в системі 3 обробки даних, запускають виконання відповідних винесених умовних переходів у пристрої 6. Таким чином, вважається, що в присутності пристрою 6 ця частина виконується коректно й, отже, захищена програма 2p є повнофункціональною.

На Фіг.92 наведений приклад спроби виконання захищеної програми 2p за відсутності пристрою 6. У цьому прикладі, в ході виконання в системі 3 обробки даних першої виконуваної частини 2pes захищеної програми 2p:

- у момент t_1 виконання команди CBC_1 умовного переходу не може викликати виконання винесеного умовного переходу bc, з огляду на відсутність пристрою 6;

- і у момент t_2 спроба передачі інформації, що дозволяє першій виконуваній частині 2pes продовжити виконання у вибраному місці не може бути успішною у зв'язку з відсутністю пристрою 6.

Таким чином, за відсутності пристрою 6 щонайменше один запит одного фрагмента першої виконуваної

UA 77187 C2

частини 2pes на запуск виконання винесеного умовного переходу у пристрої 6 не може бути коректно виконаний. Тому щонайменше ця частина не виконується коректно й, отже, захищена програма 2p не є повнофункціональною.

У попередньому описі, проілюстрованому на Фіг.90-92, даний винахід спрямований на винесення у пристрій 6 одного умовного переходу. Зрозуміло, кращий варіант здійснення винаходу може полягати в передачі у пристрій 6 серії умовних переходів, глобальні функціональні можливості яких еквівалентні ансамблю функціональних можливостей винесених умовних переходів. Виконання глобальних функціональних можливостей цієї серії винесених умовних переходів приводить до того, що системі 3 обробки даних дається інформація, що дозволяє першій виконуваній частині 2pes захищеної програми 2p продовжити своє виконання у вибраному місці.

У попередньому описі, проілюстрованому Фіг.40-92, чотири різні принципи захисту програми були пояснені в загальному вигляді незалежно одне від іншого. Спосіб захисту згідно з винаходом може бути реалізований із застосуванням принципу захисту за допомогою детектування й примусу, до якого можна додати один або декілька інших принципів захисту. В разі, коли принцип захисту за допомогою детектування й примусу доповнюється реалізацією щонайменше одного іншого принципу захисту, принцип захисту за допомогою детектування й примусу бажано доповнити принципом захисту за допомогою змінної і/або принципом захисту за допомогою перейменування, і/або принципом захисту за допомогою умовного переходу.

Якщо таким чином застосовується принцип захисту за допомогою перейменування, то він може бути доповнений, у свою чергу, принципом захисту за допомогою умовного переходу.

Відповідно до кращого варіанта реалізації принцип захисту за допомогою детектування й примусу доповнюється принципом захисту за допомогою змінної і принципом захисту за допомогою перейменування, доповненим принципом захисту за допомогою умовного переходу.

У випадку, коли на додаток до принципу захисту за допомогою детектування й примусу застосований якийсь інший принцип захисту, щоб врахувати таку комбіновану реалізацію винаходу, вищенаведений опис має містити такі модифікації:

- поняття вразливої програми має розумітися в сенсі вразливості програми відносно до описуваного принципу захисту. Так, у випадку, коли принцип захисту вже було застосовано до вразливої програми, вираз "вразлива програма" має інтерпретуватися як вираз "програма, захищена за допомогою одного або декількох уже застосованих принципів захисту";

- поняття захищеної програми має розумітися в сенсі захищеності програми відносно до описуваного принципу захисту. Так, у випадку, коли принцип захисту вже було застосовано, вираз "захищена програма" має інтерпретуватися читачем як вираз "нова версія захищеної програми";

- один або декілька виборів, зроблений (зроблені) для реалізації описуваного принципу захисту, має (мають) враховувати вибір (вибори), зроблений (зроблені) для реалізації вже застосованого (застосованих) одного або декількох принципів захисту.

Подальший опис дозволяє краще зрозуміти реалізацію способу захисту згідно з винаходом. У цьому способі згідно з винаходом здійснюють, як це уточнюється на Фіг.100:

- спочатку фазу P захисту, в ході якої вразлива програма 2N перетворюється на захищену програму 2p;

- потім фазу U використання, в ході якої захищена програма 2p використовується, причому у фазі U використання:

- у присутності пристрою 6 кожного разу, коли цього вимагає фрагмент першої виконуваної частини 2pes, виконуваної в системі 3 обробки даних, належна функціональна можливість виконується у пристрої 6 таким чином, що ця частина виконується коректно й, отже, захищена програма 2p є повнофункціональною,

- за відсутності пристрою 6, незважаючи на запит фрагмента першої виконуваної частини 2pes на виконання функціональної можливості у пристрої 6, цей запит не може бути коректно задоволений, так що принаймні ця частина не виконується коректно й, отже, захищена програма 2p не є повнофункціональною;

- і, можливо, фазу R перезавантаження, в ході якої вирішується щонайменше одне додаткове використання функціональної можливості, захищеної шляхом реалізації другого кращого варіанта реалізації принципу захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю.

Фаза P захисту може бути розкладена на дві підфази P_1 і P_2 захисту. Перша, названа вхідною, підфаза P_1 захисту реалізується незалежно від підлягаючої захисту вразливої програми 2N. Друга, названа вихідною, підфаза P_2 захисту залежить від підлягаючої захисту вразливої програми 2N. Слід зазначити, що вхідна і вихідна підфази P_1 і P_2 захисту можуть бути реалізовані двома різноманітними особами або групами. Наприклад, вхідна підфаза P_1 захисту може бути реалізована співробітником або організацією, що здійснюють розробку систем захисту програм, тоді як вихідна підфаза P_2 захисту може бути реалізована співробітником або організацією, що здійснюють розробку програм, що їх потрібно захистити. Зрозуміло, ясно, що вхідна і вихідна підфази P_1 і P_2 захисту можуть бути реалізовані й одним співробітником або однією організацією.

Вхідна підфаза P_1 захисту задіює декілька стадій S_{11}, ... S_{1i}, для кожної з яких необхідно виконати декілька задач або завдань.

Перша стадія цієї вхідної підфази P_1 захисту називається "стадією S_{11} визначень". У ході цієї стадії S_{11} визначень:

- вибираються:

-тип пристрою 6 (наприклад, як пристрій 6 можна вибрати пристрій 8 читання карт із мікрочипом і карту з мікрочипом, зв'язану з пристроєм 8 читання),

- і засоби 12, 13 передачі, призначені для застосування відповідно в системі 3 обробки даних і у пристрої 6 у фазі U використання й що здатні забезпечувати передачу даних між системою 3 обробки даних і пристроєм 6;

UA 77187 C2

UA 77187 C2

- визначаються:

- щонайменше одна характеристика виконання програми, яка може бути проконтрольована, щонайменше частково, у пристрої 6,

- щонайменше один критерій, що його слід дотримуватися щонайменше для однієї характеристики виконання програми,

- засоби 17 детектування, що їх необхідно застосовувати у пристрої 6 і які дозволяють виявити, що принаймні одна характеристика виконання програми не відповідає щонайменше одному відповідному критерію,

- і засоби 18 примусу, що їх необхідно застосовувати у пристрої 6 і які дозволяють проінформувати систему 3 обробки даних і/або модифікувати хід виконання програми, якщо хоча б одного критерію не дотримано;

- причому в разі, коли спосіб захисту згідно з винаходом задіює принцип захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю виконання програми, мають визначатися також:

- як характеристика виконання програми, яка може бути проконтрольована, - змінна для кількісного контролю використання однієї функціональної можливості програми,

- як критерій, що його необхідно дотримуватися, - щонайменше одне порогове значення, зв'язане з кожною змінною для кількісного контролю,

- і засоби поновлення, що дозволяють поновити щонайменше одну змінну для кількісного контролю;

- причому в разі, коли спосіб захисту згідно з винаходом задіює перший кращий варіант реалізації принципу захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю виконання програми, мають передбачатися також:

- щонайменше для однієї змінної для кількісного контролю - декілька відповідних порогових значень,

- і різноманітні засоби примусу, що відповідають кожному з цих порогових значень;

- причому в разі, коли спосіб захисту згідно з винаходом задіює другий кращий варіант реалізації принципу захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю виконання програми, передбачаються також засоби перезавантаження, що дозволяють щонайменше одне додаткове використання щонайменше однієї функціональної можливості програми, контрольованої за допомогою змінної для кількісного контролю;

- причому в разі, коли спосіб захисту згідно з винаходом задіює принцип захисту за допомогою детектування й примусу, що використовує як характеристику профіль використання програми, мають передбачатися також:

- як характеристика виконання програми, що може бути проконтрольована, - профіль використання програми,

- і як критерій, що його слід дотримуватися, - щонайменше одна ознака виконання програми;

- причому в разі, коли спосіб захисту згідно з винаходом задіює принцип захисту за допомогою детектування й примусу, де як ознака виконання, що її слід дотримуватися, використовується контроль зчеплення виконання, мають передбачатися також:

- набір інструкцій, інструкції зі складу якого можуть бути виконані у пристрої 6,

- набір команд інструкцій для зазначеного набору інструкцій, причому ці команди інструкцій можуть бути виконані в системі 3 обробки даних, викликаючи у пристрої 6 виконання інструкцій,

- як профіль використання - зчеплення інструкцій,

- як ознака виконання - бажане зчеплення для виконання інструкцій,

- як засоби 17 детектування - засоби, що дозволяють виявити, що зчеплення інструкцій не відповідає бажаному,

- і як засоби 18 примусу - засоби, що дозволяють інформувати систему 3 обробки даних і/або модифікувати функціонування фрагмента захищеної програми 2p, якщо зчеплення інструкцій не відповідає бажаному;

- а у випадку, коли спосіб захисту згідно з винаходом задіює кращий варіант реалізації принципу захисту за допомогою детектування й примусу, де як ознака виконання, що її слід дотримуватися, використовується контроль зчеплення виконання, мають передбачатися також:

- як набір інструкцій - набір інструкцій, з якого щонайменше деякі інструкції працюють на регістрах і використовують щонайменше один операнд для видачі результату щонайменше для однієї частини інструкцій, що працюють на регістрах:

- частина PF, що задає функціональну можливість інструкції,

- і частина, що задає бажане зчеплення для виконання інструкцій і містить бітові поля, що відповідають:

- полю СП ідентифікації інструкції,

- і для кожного операнду інструкції:

- полю CD_k прапора,

- і полю CIP_k ідентифікації, передбаченої для операнда,

- для кожного регістра, що належить до засобів експлуатації і використовується набором інструкцій, - поле CIG_v генерованої ідентифікації, в якому автоматично запам'ятовується ідентифікація останньої інструкції, що повернула свій результат у цей регістр,

- як засоби 17 детектування - засоби, що дозволяють у ході виконання інструкції для кожного операнда, якщо цього вимагає поле CD_k прапора, контролювати рівність між полем CIG_v генерованої ідентифікації, що відповідає регістру, використовуваному цим операндом, і полем CIP_k передбаченої ідентифікації початкової адреси цього операнда,

- і як засоби 18 примусу - засоби, що дозволяють модифікувати результат інструкції, якщо принаймні одна з контрольованих рівностей хибна;

- причому в разі, коли спосіб захисту згідно з винаходом задіює принцип захисту за допомогою

UA 77187 C2

UA 77187 C2

перейменування, мають передбачатися також:

- як команда запуску - елементарна команда або команда інструкції,

- як залежна функція - елементарна функція або інструкція,

- як настановний параметр, - щонайменше один аргумент для команди запуску, що відповідає, принаймні частково, інформації, переданій системою 3 обробки даних на пристрій 6, щоб викликати запуск відповідної залежної функції,

- метод перейменування настановних параметрів, що дозволяє перейменувати настановні параметри, щоб одержати команди запуску з перейменованими параметрами,

- і засоби 20 відновлення, призначені для застосування у пристрої 6 у фазі U використання і що дозволяють знову знайти залежну функцію, що її слід виконати, виходячи з перейменованого настановного параметра,

- причому в разі, коли спосіб захисту згідно з винаходом задіює варіант принципу захисту за допомогою перейменування, визначається також, щонайменше для однієї залежної функції, сімейство алгоритмічно еквівалентних залежних функцій, що викликаються командами запуску, перейменовані настановні параметри яких є різними,

- а у випадку, коли спосіб захисту згідно з винаходом, задіює ту або іншу кращу реалізацію варіанта принципу захисту за допомогою перейменування, визначається також, щонайменше для однієї залежної функції, сімейство алгоритмічно еквівалентних залежних функцій:

- за допомогою зчеплення поля шумів з інформацією, що визначає ту функціональну частину залежної функції, що виконується у пристрої 6,

- або за допомогою використання поля CII ідентифікації інструкції і поля CIP_k передбаченої ідентифікації операндів;

- причому в разі, коли спосіб захисту згідно з винаходом задіює кращий варіант принципу захисту за допомогою перейменування, мають передбачатися також:

- як метод перейменування настановних параметрів - метод кодування для кодування настановних параметрів,

- і як засоби 20 відновлення - засоби, що застосовують метод декодування для розкодування перейменованих настановних параметрів і відновлення ідентичності залежних функцій, що їх слід виконати у пристрої 6.

У ході вхідної підфази захисту за стадією S_{11} визначення випливає стадія, названа "стадія S_{12} конструювання". В ході такої стадії S_{12} конструюються засоби 12, 13 передачі й, можливо, засоби експлуатації, що відповідають визначенням стадії S_{11} визначення.

В ході цієї стадії S_{12} конструювання приступають, отже:

- до конструювання засобів 12, 13 передачі, що дозволяють, у ході фази U використання, здійснювати передачу даних між системою 3 обробки даних і пристроєм 6;

- до побудови:

- засобів експлуатації, що дозволяють пристрою 6 у фазі U використання задіяти засоби 17 детектування й засоби 18 примусу,

- і, можливо, засобів експлуатації, що дозволяють пристрою 6 у фазі U використання також задіяти засоби відновлення,

- а також, можливо, засобів експлуатації, що дозволяють пристрою 6 у фазі перезавантаження задіяти також засоби перезавантаження,

- а також, можливо, засобів експлуатації, що дозволяють пристрою 6 у фазі U використання виконувати інструкції з набору інструкцій;

- і (якщо також застосовується принцип захисту за допомогою перейменування) до побудови засобів експлуатації, що дозволяють пристрою 6 у фазі U використання задіяти також засоби відновлення.

Конструювання засобів експлуатації здійснюють звичайним чином, за допомогою пристрою розробки програм з урахуванням визначень, уведених на стадії S_{11} визначень. Подібний пристрій описано далі та проілюстровано Фіг.110.

У ході вхідної підфази P_1 захисту за стадією S_{12} конструювання може йти стадія, названа "стадією S_{13} передперсоналізації". В ході цієї стадії S_{13} передперсоналізації щонайменше засоби 13 передачі й засоби експлуатації завантажуються щонайменше в один незадіяний пристрій 60, щоб одержати щонайменше один передперсоналізований пристрій 66. Слід відзначити, що частина засобів експлуатації, бувши перенесена в передперсоналізований пристрій 66, більш недосяжна безпосередньо ззовні цього передперсоналізованого пристрою 66. Передача засобів експлуатації в незадіяний пристрій 60 може реалізуватися за допомогою адаптованого пристрою передперсоналізації, що його описано далі і проілюстровано Фіг.120. У випадку передперсоналізованого пристрою 66, що складається з карти 7 з мікрочипом і пристрою 8 її читання, передперсоналізація стосується тільки карти 7 з мікрочипом.

У ході вхідної підфази P_1 захисту, після стадії S_{11} визначень і, можливо, після стадії S_{12} конструювання, може здійснюватися також стадія, названа "стадією S_{14} виготовлення засобів". У ході цієї стадії S_{14} виготовлення засобів виробляються засоби, що дозволяють допомогти створенню захищених програм або автоматизувати захист програм. Такі засоби дозволяють:

- допомогти вибрати або автоматично вибрати в уразливій програмі 2v, що її слід захистити:

- одну або декілька характеристик виконання, що їх необхідно проконтролювати, і, можливо, один або декілька алгоритмів, які можуть розкладатися на інструкції, що їх можна винести у пристрій 6,

- фрагменти, які можуть бути змінені,

UA 77187 C2

UA 77187 C2

- якщо також застосовується принцип захисту за допомогою змінної, - одну або декілька змінних, які можуть бути винесені у пристрій 6,

- якщо також застосовується принцип захисту за допомогою перейменування, - один або декілька алгоритмів, які можуть розкладатися на залежні функції, що їх можна винести у пристрій 6 і для яких настановні параметри команд запуску можуть бути перейменовані,

- і, якщо також застосовується принцип захисту за допомогою умовного переходу, - один або декілька умовних переходів, функціональна можливість яких може бути винесена в пристрій 6;

- і, можливо, допомогти створити захищені програми або автоматизувати захист програм.

Ці різноманітні засоби можуть реалізуватися незалежно або в поєднанні, причому кожен засіб може приймати різноманітні форми, наприклад являти собою перепроцесор, асемблер, компілятор тощо.

За вхідною підфазою P_1 захисту випливає вихідна підфаза P_2 захисту, що залежить від підлягаючої захисту вразливої програми 2v. Ця вихідна підфаза P_2 захисту також передбачає декілька стадій. Перша стадія, що відповідає реалізації принципу захисту за допомогою детектування й примусу, називається "стадією S_{21} створення". В ході цієї стадії S_{21} створення використовується вибір, зроблений на стадії S_{11} визначень. За допомогою цього вибору і, можливо, засобів, сконструйованих на стадії S_{14} виготовлення засобів, захищена програма 2p створюється:

- за допомогою вибору серед характеристик виконання, які можуть бути проконтрольовані, щонайменше однієї характеристики виконання контрольованої програми;

- за допомогою вибору щонайменше одного критерію, який має виконуватися, щонайменше для однієї вибраної характеристики виконання програми;

- за допомогою вибору щонайменше одного алгоритму, який у ході виконання вразливої програми 2v використовує щонайменше один операнд і дозволяє одержати щонайменше один результат і для якого слід контролювати щонайменше одну характеристику виконання вибраної програми,

- за допомогою вибору щонайменше одного фрагмента коду 2vs уразливої програми, що містить щонайменше один вибраний алгоритм,

- за допомогою створення вихідного коду 2ps захищеної програми на основі коду вразливої програми 2v модифікацією щонайменше одного вибраного фрагмента коду 2vs уразливої програми, щоб одержати щонайменше один модифікований фрагмент коду 2ps захищеної програми, причому ця модифікація така, що:

- в ході виконання захищеної програми 2p перша виконувана частина 2pes виконується в системі 3 обробки даних, а друга виконувана частина 2peu виконується у пристрої 6, отриманому з незадіяного пристрою 60 після завантаження інформації,

- друга виконувана частина 2peu виконує щонайменше функціональну можливість щонайменше одного вибраного алгоритму,

- і в ході виконання захищеної програми 2p щонайменше одна вибрана характеристика виконання контролюється за допомогою другої виконуваної частини 2peu і недотримання критерію приводить до модифікації виконання захищеної програми 2p;

- і за допомогою створення:

- першої частини 2pos об'єктного коду захищеної програми 2p, причому перша частина 2pos об'єктного коду така, що в ході виконання захищеної програми 2p реалізується перша виконувана частина 2pes, яка виконується в системі 3 обробки даних, і щонайменше частина якої враховує, що контролюється щонайменше одна характеристика виконання вибраної програми,

- і другої частини 2pou об'єктного коду захищеної програми 2p, що містить засоби експлуатації, що дозволяють застосовувати також засоби 17 детектування і засоби 18 примусу, причому друга частина 2pou об'єктного коду така, що після завантаження в незадіяний пристрій 60, у ході виконання захищеної програми 2p реалізується друга виконувана частина 2peu, за допомогою якої контролюється щонайменше одна характеристика виконання програми, і за допомогою якої недотримання критерію приводить до модифікації виконання захищеної програми 2p.

Зрозуміло, принцип захисту за допомогою детектування й примусу згідно з винаходом може бути застосований безпосередньо в ході розробки нової програми без необхідності попередньої реалізації вразливої програми 2v. У цьому випадку безпосередньо одержують захищену програму 2p.

Для реалізації принципу захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю виконання програми, захищена програма 2p модифікується:

- за допомогою вибору як характеристики виконання контрольованої програми щонайменше однієї змінної для кількісного контролю використання однієї функціональної можливості програми;

- за допомогою вибору:

- щонайменше однієї функціональної можливості захищеної програми 2p, використання якої можна проконтролювати з використанням змінної для кількісного контролю,

- щонайменше однієї змінної для кількісного контролю, яка служить як кількісна характеристика використання згаданої функціональної можливості,

- щонайменше одного порогового значення, зв'язаного з вибраною змінною для кількісного контролю і відповідного межі використання згаданої функціональної можливості,

- і щонайменше одного методу поновлення значення змінної для кількісного контролю відповідно до використання згаданої функціональної можливості;

- і за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p змінна для кількісного контролю

UA 77187 C2

UA 77187 C2

поновлюється за допомогою другої виконуваної частини 2peu, залежно від використання згаданої функціональної можливості, і враховується щонайменше одне перевищення порогового значення.

Для реалізації першого кращого варіанта принципу захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю, захищена програма 2p модифікується:

- за допомогою вибору у вихідному коді 2ps захищеної програми щонайменше однієї змінної для кількісного контролю, з якою мають зв'язуватися декілька порогових значень, що відповідають різноманітним межам використання функціональної можливості;

- за допомогою вибору щонайменше двох порогових значень, зв'язаних з вибраною змінною для кількісного контролю;

- і за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p перевищення різноманітних порогових значень враховуються за допомогою другої виконуваної частини 2peu різним способом.

Для реалізації другого кращого варіанта принципу захисту за допомогою детектування й примусу, що використовує як характеристику змінну для кількісного контролю, захищена програма 2p модифікується:

- за допомогою вибору у вихідному коді 2ps захищеної програми щонайменше однієї змінної для кількісного контролю, що дозволяє обмежити використання функціональної можливості, для якої має існувати можливість дозволити щонайменше одне додаткове використання;

- і за допомогою модифікації щонайменше одного вибраного фрагмента, причому ця модифікація така, що у фазі, названій фазою перезавантаження, щонайменше одне додаткове використання щонайменше однієї функціональної можливості, що відповідає одній вибраній змінній для кількісного контролю, може бути дозволене.

Для реалізації принципу захисту за допомогою детектування й примусу, що використовує як характеристику профіль використання програми, захищена програма 2p модифікується:

- за допомогою вибору як характеристики виконання контрольованої програми щонайменше профілю використання програми;

- за допомогою вибору щонайменше однієї ознаки виконання програми, що її має дотримуватися щонайменше один профіль використання;

- і за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p друга виконувана частина 2peu дотримується всіх вибраних ознак виконання.

Для реалізації принципу захисту за допомогою детектування й примусу, де як ознака виконання, що її слід дотримуватися, використовується контроль зчеплення виконання, захищена програма 2p модифікується:

- за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми, причому ця модифікація така, що:

- щонайменше один вибраний алгоритм розкладається таким чином, що в ході виконання захищеної програми 2p цей алгоритм виконується за допомогою другої виконуваної частини 2peu, із застосуванням інструкцій,

- щонайменше для одного вибраного алгоритму команди інструкцій інтегруються у вихідний код 2ps захищеної програми таким чином, що в ході виконання захищеної програми 2p кожна команда інструкції виконується за допомогою першої виконуваної частини 2pes і викликає у пристрої 6 виконання інструкції за допомогою другої виконуваної частини 2peu,

- впорядкування команд інструкцій вибирається серед ансамблю впорядкувань, що дозволяють виконання захищеної програми 2p,

- і визначене зчеплення, що його мають дотримуватися щонайменше деякі з інструкцій у ході їхнього виконання у пристрої 6.

У ході вихідної підфази P_2 захисту, в разі, коли застосований щонайменше один інший принцип захисту, на додаток до принципу захисту за допомогою детектування й примусу, реалізується "стадія S_{22} модифікації". В ході цієї стадії S_{22} модифікації використовуються визначення, введені на стадії S_{11} визначень. З використанням цих визначень і, можливо, засобів, сконструйованих на стадії S_{14} виготовлення засобів, захищена програма 2p модифікується таким чином, щоб дозволити реалізувати принципи захисту відповідно до одного з компонувань, визначених вище.

Коли застосовується принцип захисту за допомогою змінної, то захищена програма 2p модифікується:

- за допомогою вибору щонайменше однієї змінної, використовуваної щонайменше в одному вибраному алгоритмі, яка в ході виконання захищеної програми 2p частково визначає стан захищеної програми 2p,

- за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p щонайменше одна вибрана змінна або одна копія вибраної змінної знаходиться у пристрої 6,

- і за допомогою створення:

- першої частини 2pos об'єктного коду захищеної програми 2p, причому ця перша частина 2pos об'єктного коду така, що в ході виконання захищеної програми 2p щонайменше один фрагмент першої виконуваної частини 2pes враховує, що принаймні одна змінна або щонайменше одна копія змінної знаходиться у пристрої 6,

- і другої частини 2pou об'єктного коду захищеної програми 2p, причому ця друга частина 2pou об'єктного коду така, що після завантаження в пристрій 6 і в ході виконання захищеної програми 2p реалізується друга виконувана частина 2peu, за допомогою якої щонайменше одна вибрана змінна або щонайменше одна копія вибраної змінної також знаходиться у пристрої 6.

UA 77187 C2

UA 77187 C2

Коли застосовується принцип захисту за допомогою перейменування, захищена програма 2p модифікується:

- за допомогою вибору у вихідному коді 2ps захищеної програми команд запуску;

- за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми за допомогою перейменування настановних параметрів вибраних команд запуску, щоб приховати ідентичність відповідних залежних функцій;

- і за допомогою створення:

- першої частини 2pos об'єктного коду захищеної програми 2p, причому ця перша частина 2pos об'єктного коду така, що в ході виконання захищеної програми 2p виконуються команди запуску з перейменованими настановними параметрами,

- і другої частини 2pou об'єктного коду захищеної програми 2p, що містить засоби експлуатації, що використовують також засоби 20 відновлення, причому ця друга частина 2pou об'єктного коду така, що після завантаження у пристрій 6, у ході виконання захищеної програми 2p ідентичність залежних функцій, виконання яких запускається першою виконуваною частиною 2pes, відновлюється за допомогою другої виконуваної частини 2peu, а залежні функції виконуються за допомогою другої виконуваної частини 2peu.

Для реалізації варіанта принципу захисту перейменуванням захищена програма 2p модифікується:

- за допомогою вибору у вихідному коді 2ps захищеної програми щонайменше однієї команди запуску з перейменованими настановними параметрами;

- і за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми за допомогою заміни щонайменше перейменованого настановного параметра команди запуску з вибраним настановним параметром на інший перейменований настановний параметр, що викликає запуск залежної функції, з того ж сімейства.

Коли застосовується принцип захисту за допомогою умовного переходу, захищена програма 2p модифікується:

- за допомогою вибору у вихідному коді 2ps захищеної програми щонайменше одного умовного переходу, виконуваного щонайменше в одному вибраному алгоритмі;

- за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p функціональна можливість щонайменше одного вибраного умовного переходу виконується за допомогою другої виконуваної частини 2peu у пристрої 6;

- і за допомогою створення:

- першої частини 2pos об'єктного коду захищеної програми 2p, причому ця перша частина 2pos об'єктного коду така, що в ході виконання захищеної програми 2p функціональна можливість щонайменше одного вибраного умовного переходу виконується у пристрої 6,

- і другої частини 2pou об'єктного коду захищеної програми 2p, причому ця друга частина 2pou об'єктного коду така, що після завантаження у пристрій 6, у ході виконання захищеної програми 2p реалізується друга виконувана частина 2peu, за допомогою якої виконується функціональна можливість щонайменше одного вибраного умовного переходу.

Для кращої реалізації принципу захисту за допомогою умовного переходу захищена програма 2p модифікується:

- за допомогою вибору у вихідному коді 2ps захищеної програми щонайменше однієї серії вибраних умовних переходів;

- за допомогою модифікації щонайменше одного вибраного фрагмента коду 2ps захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми 2p глобальна функціональна можливість щонайменше однієї вибраної серії умовних переходів виконується за допомогою другої виконуваної частини 2peu у пристрої 6;

- і за допомогою створення:

- першої частини 2pos об'єктного коду захищеної програми 2p, причому ця перша частина 2pos об'єктного коду така, що в ході виконання захищеної програми 2p функціональна можливість щонайменше однієї вибраної серії умовних переходів виконується у пристрої 6,

- і другої частини 2pou об'єктного коду захищеної програми 2p, причому ця друга частина 2pou об'єктного коду така, що після завантаження у пристрій 6, у ході виконання захищеної програми 2p реалізується друга виконувана частина 2peu, за допомогою якої виконується глобальна функціональна можливість щонайменше однієї вибраної серії умовних переходів.

Зрозуміло, принципи захисту згідно з винаходом можуть застосовуватися безпосередньо в ході розробки нової програми без попередньої реалізації проміжних уразливих програм. Таким чином, стадії S_{21} створення і S_{22} модифікації можуть бути здійснені одночасно, щоб відразу одержати захищену програму 2p.

Під час вихідної під фази P_2 захисту після стадії S_{21} створення захищеної програми 2p і, можливо, після стадії S_{22} модифікації реалізується стадія, названа "стадією S_{23} персоналізації". В ході цієї стадії S_{23} персоналізації друга частина 2pou об'єктного коду, що, можливо, містить засоби експлуатації, завантажується щонайменше в один незадіяний пристрій 60, щоб одержати щонайменше один пристрій 6. Альтернативно, частина другої частини 2pou об'єктного коду, що, можливо, містить засоби експлуатації, завантажується щонайменше в один передперсоналізований пристрій 66, щоб одержати щонайменше один пристрій 6. Завантаження цієї персоналізуючої інформації дозволяє зробити працездатним щонайменше один пристрій 6. Слід зазначити, що частина цієї інформації, бувши перенесеною у пристрій 6, недосяжна безпосередньо ззовні цього пристрою 6. Передача інформації персоналізації в незадіяний пристрій 60 або в передперсоналізований

UA 77187 C2

пристрій 66 може бути реалізована за допомогою адаптованого пристрою персоналізації, що його описано далі і проілюстровано Фіг.150. У разі пристрою 6, що складається з карти 7 з мікрочипом і пристрою 8 її читання, персоналізація стосується тільки карти 7 з мікрочипом.

Різноманітні технічні засоби для реалізації фази P захисту, які далі будуть описані докладніше, проілюстровані Фіг.110, 120, 130, 140 і 150.

На Фіг.110 наведений приклад реалізації системи 25, що дозволяє реалізувати стадію S_{12} конструювання з урахуванням визначень, уведених на стадії S_{11} визначень, і в ході якої конструюються засоби 12, 13 передачі й, можливо, засоби експлуатації, призначені для пристрою 6. Подібна система 25 містить пристрій розробки програм або робочу станцію, що звичайно представляє собою комп'ютер, який має системний блок, монітор, периферійні пристрої типу клавіатури і миші і на якому встановлені такі програми: файлові редактори, асемблери, перепроцесори, компілятори, інтерпретатори, налагоджувані і редактори зв'язків.

На Фіг.120 наведений приклад реалізації пристрою 30 перед-персоналізації, що дозволяє завантажити, щонайменше частково, засоби 13 передачі та/або засоби експлуатації щонайменше в один незадіяний пристрій 60, щоб одержати щонайменше передперсоналізований пристрій 66. Пристрій 30 передперсоналізації містить засіб 31 читання-запису, що дозволяє електрично передперсоналізувати незадіяний пристрій 60, щоб одержати передперсоналізований пристрій 66, у який завантажені засоби 13 передачі і/або засоби експлуатації. Пристрій 30 передперсоналізації може також містити фізичні засоби 32 передперсоналізації незадіяного пристрою 60, що являють собою, наприклад, принтер. У випадку, якщо пристрій 6 складається з карти 7 з мікрочипом і пристрою 8 її читання, передперсоналізація стосується звичайно тільки карти 7 з мікрочипом.

На Фіг.130 наведений приклад реалізації системи 35, що дозволяє здійснити виготовлення засобів, призначених для використання при створенні захищених програм або автоматизації захисту програм. Подібна система 35 містить пристрій розробки програм або робочої станції, що звичайно представляє собою комп'ютер, який містить системний блок, монітор, периферійні пристрої типу клавіатури і миші і на якому є такі програми: файлові редактори, асемблери, перепроцесори, компілятори, інтерпретатори, налагоджувачі і редактори зв'язків.

На Фіг.140 наведений приклад реалізації системи 40, що дозволяє безпосередньо одержати захищену програму 2p або модифікувати вразливу програму 2v з метою одержати захищену програму 2p. Подібна система 40 включає пристрій розробки програми або робочу станцію, що звичайно представляє собою комп'ютер, що має системний блок, монітор, периферійні пристрої типу клавіатури і миші і на якому встановлені такі програми: файлові редактори, асемблери, перепроцесори, компілятори, інтерпретатори, налагоджувані і редактори зв'язків, а також засоби, що допомагають при створенні захищених програм або автоматизації захисту програм.

На Фіг.150 наведений приклад реалізації пристрою 45 персоналізації, що дозволяє завантажити другу частину 2pou об'єктного коду щонайменше в один незадіяний пристрій 60, щоб одержати щонайменше один пристрій 6, або частину другої частини 2pou об'єктного коду щонайменше в один передперсоналізований пристрій 66, щоб одержати щонайменше один пристрій 6. Цей пристрій 45 персоналізації містить засіб 46 читання-запису, що дозволяє електрично персоналізувати щонайменше один незадіяний пристрій 60 або щонайменше один передперсоналізований пристрій 66, щоб одержати щонайменше один пристрій 6. По завершенні цієї персоналізації пристрій 6 містить інформацію, необхідну для виконання захищеної програми 2p. Пристрій 45 персоналізації може також містити фізичні засоби 47 персоналізації щонайменше для одного пристрою 6, що являють собою, наприклад, принтер. У випадку, якщо пристрій 6 складається з карти 7 з мікрочипом і пристрою 8 її читання, персоналізація стосується звичайно тільки карти 7 з мікрочипом.

Спосіб захисту згідно з винаходом може реалізовуватися з додатковими вдосконаленнями.

- Можна передбачити спільне використання множини пристроїв обробки і зберігання, між якими розподілена друга частина 2pou об'єктного коду захищеної програми 2p таким чином, що їхнє спільне виконання дозволяє виконати захищену програму 2p, відсутність же щонайменше одного з цих пристроїв обробки і зберігання перешкоджає використанню захищеної програми 2p.

- Аналогічно, після стадії S_{13} передперсоналізації й у ході стадії S23 персоналізації частина другої частини 2pou об'єктного коду, необхідна для перетворення передперсоналізованого пристрою 66 на пристрій 6, може міститися у пристрої обробки й зберігання, використовуваному пристроєм 45 персоналізації, щоб обмежити доступ до цієї частини другої частини 2pou об'єктного коду. Зрозуміло, ця частина другої частини 2pou об'єктного коду може бути розподілена між декількома пристроями обробки і зберігання таким чином, щоб ця частина другої частини 2pou об'єктного коду була досяжною тільки під час спільного використання цих пристроїв обробки і зберігання.

Формула винаходу

1. Спосіб захисту, на основі щонайменше одного незадіяного пристрою (60), що містить принаймні засоби (15) запам'ятовування і засоби (16) обробки, від неавторизованого використання уразливої програми (2v), яка функціонує на системі (3) обробки даних, який полягає в тому, що:

а у фазі (P) захисту:

• визначають:

– щонайменше одну характеристику виконання програми, що може бути проконтрольована, щонайменше частково, у пристрої (6),

– щонайменше один критерій, що має виконуватися щонайменше для однієї характеристики виконання

UA 77187 C2

програми,

– засоби (17) детектування, що їх слід застосовувати у пристрої (6) і які дозволяють виявити, що принаймні одна характеристика виконання програми не відповідає щонайменше одному відповідному критерію,

– і засоби (18) примусу, що їх слід застосовувати у пристрої (6) і які дозволяють проінформувати систему (3) обробки даних і/або модифікувати виконання програми, поки не дотримано хоча б одного критерію;

• конструюють засоби експлуатації, що дозволяють пристрою (6) задіяти засоби (17) детектування й засоби (18) примусу;

створюють захищену програму (2p):

– за допомогою вибору щонайменше однієї характеристики виконання контрольованої програми з характеристик виконання, які можуть бути проконтрольовані,

– за допомогою вибору щонайменше одного критерію, який має виконуватися щонайменше для однієї вибраної характеристики виконання програми,

– за допомогою вибору щонайменше одного алгоритму, який у ході виконання уразливої програми (2v) використовує щонайменше один операнд і дозволяє одержати щонайменше один результат і для якого слід контролювати щонайменше одну характеристику виконання вибраної програми,

– за допомогою вибору щонайменше одного фрагмента коду (2vs) уразливої програми, що містить щонайменше один вибраний алгоритм,

– за допомогою створення вихідного коду (2ps) захищеної програми на основі коду уразливої програми (2v) модифікацією щонайменше одного вибраного фрагмента коду (2vs) уразливої програми, щоб одержати щонайменше один модифікований фрагмент коду (2ps) захищеної програми, причому ця модифікація така, що:

Ш в ході виконання захищеної програми (2p) перша виконувана частина (2pes) виконується в системі (3) обробки даних, а друга виконувана частина (2peu) виконується у пристрої (6), отриманому з незадіяного пристрою (60) після завантаження інформації,

Ш друга виконувана частина (2peu) виконує щонайменше функціональну можливість щонайменше одного вибраного алгоритму,

Ш і в ході виконання захищеної програми (2p) щонайменше одна вибрана характеристика виконання контролюється за допомогою другої виконуваної частини (2peu), і недотримання критерію приводить до модифікації виконання захищеної програми (2p);

– і за допомогою створення:

Ш першої частини (2pos) об'єктного коду захищеної програми (2p), причому перша частина (2pos) об'єктного коду така, що в ході виконання захищеної програми (2p) реалізується перша виконувана частина (2pes), яка виконується в системі обробки даних (3), і щонайменше частина якого враховує, що контролюється щонайменше одна характеристика виконання вибраної програми,

Ш і другої частини (2pou) об'єктного коду захищеної програми (2p), що містить засоби експлуатації, що дозволяють застосовувати також засоби (17) детектування і засоби (18) примусу, причому друга частина (2pou) об'єктного коду така, що після завантаження в незадіяний пристрій (60), у ході виконання захищеної програми (2p) реалізується друга виконувана частина (2peu), за допомогою якої контролюється щонайменше одна характеристика виконання програми і за допомогою якої недотримання критерію приводить до модифікації виконання захищеної програми (2p),

• і завантажують другу частину (2pou) об'єктного коду в незадіяний пристрій (60) з одержанням пристрою (6),

а а в фазі (U) використання, в ході якої відбувається виконання захищеної програми (2p):

• за наявності пристрою (6):

– якщо всіх критеріїв, що відповідають усім контрольованим характеристикам виконання всіх модифікованих фрагментів захищеної програми (2p), дотримано, допускають номінальне функціонування зазначених фрагментів захищеної програми (2p) й, отже, номінальне функціонування захищеної програми (2p),

– а якщо принаймні одного з критеріїв, що відповідають контрольованій характеристиці виконання одного фрагмента захищеної програми (2p), не дотримано, інформують систему (3) обробки даних про згадане недотримання і/або модифікують функціонування фрагмента захищеної програми (2p) таким чином, щоб функціонування захищеної програми (2p) було змінено;

• тоді як за відсутності пристрою (6), незважаючи на запит фрагмента першої виконуваної частини (2pes) на запуск виконання у пристрої (6) функціональної можливості вибраного алгоритму, не забезпечується можливість коректно відповісти на цей запит, так що принаймні вказаний фрагмент не виконується коректно й, отже, захищена програма (2p) не є повнофункціональною.

2. Спосіб за п. 1, який **відрізняється** тим, що:

щоб обмежити використання захищеної програми (2p),

а у фазі (P) захисту:

• визначають:

– як характеристику виконання програми, яка може бути проконтрольована, - змінну для кількісного контролю використання однієї функціональної можливості програми,

– як критерій, що його слід дотримуватися, - щонайменше одне порогове значення, зв'язане з кожною змінною для кількісного контролю,

– і засоби поновлення, що дозволяють поновити щонайменше одну змінну для кількісного контролю;

• конструюють засоби експлуатації, що дозволяють пристрою (6) застосовувати засоби поновлення;

• і модифікують захищену програму (2p):

– за допомогою вибору як характеристики виконання контрольованої програми щонайменше однієї змінної

UA 77187 C2

для кількісного контролю використання однієї функціональної можливості програми,

– за допомогою вибору:

Ш щонайменше однієї функціональної можливості захищеної програми (2p), використання якої можна проконтролювати з використанням змінної для кількісного контролю,

Ш щонайменше однієї змінної для кількісного контролю, яка служить для кількісної характеристики використання згаданої функціональної можливості,

Ш щонайменше одного порогового значення, зв'язаного з вибраною змінною для кількісного контролю і відповідного межі використання згаданої функціональної можливості,

Ш і щонайменше одного методу поновлення значення вказаної змінної для кількісного контролю залежно від використання вказаної функціональної можливості,

– і за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми, причому згадана модифікація така, що в ході виконання захищеної програми (2p) змінна для кількісного контролю поновлюється за допомогою другої виконуваної частини (2peu), залежно від використання зазначеної функціональної можливості, й ураховується щонайменше одне перевищення порогового значення,

а а у фазі (U) використання, у присутності пристрою (6), у випадку, коли виявлено щонайменше одне перевищення порогового значення, що відповідає щонайменше одній межі використання, інформують про це систему (3) обробки даних і/або модифікують функціонування фрагмента захищеної програми (2p) таким чином, щоб функціонування захищеної програми (2p) було змінено.

3. Спосіб за п. 2, який **відрізняється** тим, що:

а у фазі (P) захисту:

• визначають:

– декілька відповідних порогових значень щонайменше для однієї змінної для кількісного контролю,

– і різноманітні засоби примусу, що відповідають кожному зі згаданих порогів;

• і модифікують захищену програму (2p):

– за допомогою вибору у вихідному коді (2ps) захищеної програми щонайменше однієї змінної для кількісного контролю, з якою мають зв'язуватися декілька порогових значень, що відповідають різноманітним межам використання функціональної можливості,

– за допомогою вибору щонайменше двох порогових значень, зв'язаних з вибраною змінною для кількісного контролю,

– і за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми, причому зазначена модифікація така, що в ході виконання захищеної програми (2p) перевищення різних порогових значень враховуються за допомогою другої виконуваної частини (2peu) різними способами,

а а у фазі (U) використання:

• за наявності пристрою (6):

– у випадку, коли виявлено перевищення першого порогового значення, дають команду захищеній програмі (2p) не використовувати надалі відповідну функціональну можливість,

– а у випадку, коли виявлено перевищення другого порогового значення, роблять нездійсненною відповідну функціональну можливість і/або щонайменше частину захищеної програми (2p).

4. Спосіб за п. 2 або 3, який **відрізняється** тим, що:

а у фазі (P) захисту:

• визначають засоби перезавантаження, що дозволяють щонайменше одне додаткове використання щонайменше однієї функціональної можливості програми, контрольованої за допомогою змінної для кількісного контролю;

• конструюють засоби експлуатації, що дозволяють також пристрою (6) задіяти засоби перезавантаження;

• і модифікують захищену програму (2p):

– за допомогою вибору у вихідному коді (2ps) захищеної програми щонайменше однієї змінної для кількісного контролю, що дозволяє обмежити використання однієї функціональної можливості, для якої існує можливість дозволу, щонайменше на одне додаткове використання,

– і за допомогою модифікації щонайменше одного вибраного фрагмента, причому згадана модифікація така, що у фазі перезавантаження щонайменше одне додаткове використання щонайменше однієї функціональної можливості, що відповідає одній вибраній змінній для кількісного контролю, може бути дозволено,

а а у фазі перезавантаження:

• поновлюють щонайменше одну вибрану змінну для кількісного контролю і щонайменше одне відповідне порогове значення так, щоб дозволити щонайменше одне додаткове використання функціональної можливості.

5. Спосіб за п. 1, який **відрізняється** тим, що:

а у фазі (P) захисту:

• визначають:

– як характеристику виконання програми, яка може бути проконтрольована, - профіль використання програми,

– і як критерій, що його слід дотримуватися, - щонайменше одну ознаку виконання програми;

• і модифікують захищену програму (2p):

– за допомогою вибору як характеристики виконання контрольованої програми щонайменше одного профілю використання програми,

– за допомогою вибору щонайменше однієї ознаки виконання програми, що її має дотримуватися щонайменше один профіль використання,

UA 77187 C2

UA 77187 C2

– і за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми, причому згадана модифікація така, що в ході виконання захищеної програми (2p) друга виконувана частина (2peu) дотримується всіх вибраних ознак виконання,

а а у фазі (U) використання, за наявності пристрою (6) у випадку, якщо виявлено, що не дотримано хоча б однієї ознаки виконання, інформують про це систему (3) обробки даних і/або модифікують функціонування частини захищеної програми (2p) так, щоб функціонування захищеної програми (2p) було змінено.

6. Спосіб за п. 5, який **відрізняється** тим, що:

а у фазі (P) захисту:

• визначають:

– набір інструкцій, інструкції зі складу якого можуть бути виконані у пристрої (6),

– набір команд інструкцій для згаданого набору інструкцій, причому команди інструкцій можуть бути виконані в системі (3) обробки даних, викликаючи у пристрої (6) виконання інструкцій,

– як профіль використання – зчеплення інструкцій,

– як ознаку виконання – бажане зчеплення для виконання інструкцій,

– як засоби (17) детектування – засоби, що дозволяють виявити, що зчеплення інструкцій не відповідає бажаному,

– як засоби (18) примусу – засоби, що дозволяють проінформувати систему (3) обробки даних і/або модифікувати функціонування фрагмента захищеної програми (2p), якщо зчеплення інструкцій не відповідає бажаному;

• конструюють засоби експлуатації, що дозволяють пристрою (6) виконувати інструкції з набору інструкцій, причому виконання зазначених інструкцій викликається виконанням команд інструкцій у системі (3) обробки даних;

• і модифікують захищену програму (2p):

– за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми:

Ш за допомогою перетворення елементарних функцій на інструкції,

Ш за допомогою задання зчеплення, що його мають дотримуватися щонайменше деякі з інструкцій під час їхнього виконання у пристрої (6),

Ш і за допомогою перетворення елементарних команд на команди інструкцій, що відповідають використовуваним інструкціям,

а а у фазі (U) використання, за наявності пристрою (6), у випадку, якщо виявлено, що зчеплення виконуваних у пристрої (6) інструкцій не відповідає бажаному, інформують про це систему (3) обробки даних і/або модифікують функціонування фрагмента захищеної програми (2p) так, щоб функціонування захищеної програми (2p) було змінено.

7. Спосіб за п. 6, який **відрізняється** тим, що:

а у фазі (P) захисту:

• визначають:

– як набір інструкцій – набір інструкцій, у якому щонайменше деякі інструкції працюють на регістрах, і використовують щонайменше один операнд для видачі результату,

– щонайменше для частини інструкцій, що працюють на регістрах:

Ш частину (PF), що задає функціональну можливість інструкції,

Ш і частину, що задає бажане зчеплення для виконання інструкцій і містить бітові поля, що відповідають:

- полю (CII) ідентифікації інструкції,

- і для кожного операнда інструкції:

* полю (CD_k) прапора,

* і полю (CIP_k) ідентифікації, передбаченої для операнда,

– для кожного регістра, що належить до засобів експлуатації і використовується набором інструкцій, - поле (CIG_v) передбаченої ідентифікації, в якому автоматично запам'ятовується ідентифікація останньої інструкції, що повернула свій результат у вказаний регістр,

– як засоби (17) детектування – засоби, що дозволяють під час виконання інструкції для кожного операнда, коли цього вимагає поле (CD_k) прапора, контролювати рівність між полем (CIG_v) генерованої ідентифікації, що відповідає регістру, використовуваному згаданим операндом, і полем (CIP_k) передбаченої ідентифікації початкової адреси цього операнда,

– і як засоби (18) примусу – засоби, що дозволяють модифікувати результат виконання інструкцій, якщо принаймні одна з контрольованих рівностей хибна.

8. Спосіб за будь-яким з пп. 1-7, який **відрізняється** тим, що:

а у фазі (P) захисту:

• модифікують захищену програму (2p):

– за допомогою вибору щонайменше однієї змінної, використовуваної щонайменше в одному вибраному алгоритмі, яка в ході виконання захищеної програми (2p) частково визначає стан захищеної програми (2p),

– за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми, причому ця модифікація така, що в ході виконання захищеної програми (2p) щонайменше одна вибрана змінна або одна копія вибраної змінної знаходиться у пристрої (6),

– і за допомогою створення:

Ш першої частини (2pos) об'єктного коду захищеної програми (2p), причому ця перша частина (2pos) об'єктного коду така, що в ході виконання захищеної програми (2p) щонайменше один фрагмент першої

UA 77187 C2

виконуваної частини (2pes) враховує, що принаймні одна змінна або щонайменше одна копія змінної знаходиться у пристрої (6),

Ш і другої частини (2pou) об'єктного коду захищеної програми (2p), причому ця друга частина (2pou) об'єктного коду така, що після завантаження у пристрій (6) і в ході виконання захищеної програми (2p) реалізується друга виконувана частина (2peu), за допомогою якої щонайменше одна вибрана змінна або щонайменше одна копія вибраної змінної також знаходиться у пристрої (6),

а а у фазі (U) використання:

• за наявності пристрою (6), кожного разу, коли цього вимагає фрагмент першої виконуваної частини (2pes), використовують змінну або копію змінної, що знаходиться у пристрої (6) так, щоб указаний фрагмент виконувався коректно й, отже, захищена програма (2p) була повнофункціональною,

• а за відсутності пристрою, незважаючи на запит фрагмента першої виконуваної частини (2pes) на використання змінної або копії змінної, що знаходиться у пристрої (6), не забезпечується можливість коректної відповіді на вказаний запит, так що принаймні вказаний фрагмент не виконується коректно й, отже, захищена програма (2p) не є повнофункціональною.

9. Спосіб за п. 6, який **відрізняється** тим, що:

а у фазі (P) захисту:

• визначають:

– як команду запуску – команду інструкції,

– як залежну функцію – інструкцію,

– як установний параметр, - принаймні один аргумент для команди запуску, що відповідає, щонайменше частково, інформації, переданій системою (3) обробки даних на пристрій (6), щоб викликати запуск відповідної залежної функції,

– метод перейменування установних параметрів, що дозволяє перейменувати установні параметри, щоб одержати команди запуску з перейменованими установними параметрами,

– і засоби (20) відновлення, що призначені для застосування у пристрої (6) у фазі (U) використання і що дозволяють знайти залежну функцію, що її слід виконати, виходячи з перейменованого установного параметра;

• конструюють засоби експлуатації, дозволяють пристрою (6) задіяти засоби відновлення;

• і модифікують захищену програму (2p):

– за допомогою вибору у вихідному коді (2ps) захищеної програми команд запуску,

– за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми за допомогою перейменування установних параметрів вибраних команд запуску, щоб приховати ідентичність відповідних залежних функцій,

– і за допомогою створення:

Ш першої частини (2pos) об'єктного коду захищеної програми (2p), причому перша частина (2pos) об'єктного коду така, що в ході виконання захищеної програми (2p) виконуються команди запуску з перейменованими установними параметрами,

Ш і другої частини (2pou) об'єктного коду захищеної програми (2p), що містить засоби експлуатації, які використовують також засоби (20) відновлення, причому друга частина (2pou) об'єктного коду така, що після завантаження у пристрій (6), у ході виконання захищеної програми (2p) ідентичність залежних функцій, виконання яких викликано першою виконуваною частиною (2pes), відновлюється за допомогою другої виконуваної частини (2peu), а залежні функції виконуються за допомогою другої виконуваної частини (2peu),

а а у фазі (U) використання:

• за наявності пристрою (6), кожного разу, коли цього вимагає команда запуску з перейменованими установними параметрами, що міститься у фрагменті першої виконуваної частини (2pes), відновлюють у пристрої (6) ідентичність відповідної залежної функції і виконують її так, щоб указаний фрагмент виконувався коректно й, отже, захищена програма (2p) була повнофункціональною;

• тоді як за відсутності пристрою (6), незважаючи на запит фрагмента першої виконуваної частини (2pes) на запуск виконання у пристрої (6) залежної функції, не забезпечується можливість коректної відповіді на вказаний запит, так що принаймні вказаний фрагмент не виконується коректно й, отже, захищена програма (2p) не є повнофункціональною.

10. Спосіб за п. 9, який **відрізняється** тим, що:

а у фазі (P) захисту:

• визначають, щонайменше для однієї залежної функції, сімейство алгоритмічно еквівалентних залежних функцій, що викликаються командами запуску, перейменовані установні параметри яких є різними;

• і модифікують захищену програму (2p):

– за допомогою вибору у вихідному коді (2ps) захищеної програми щонайменше однієї команди запуску з перейменованими установними параметрами,

– і за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми за допомогою заміни щонайменше перейменованого установного параметра команди запуску з вибраним установним параметром на інший перейменований установний параметр, що викликає запуск залежної функції з того ж сімейства.

11. Спосіб за п. 10, який **відрізняється** тим, що включає:

а у фазі (P) захисту визначення, щонайменше для однієї функції, сімейства алгоритмічно еквівалентних залежних функцій:

– за допомогою зчеплення поля шумів з інформацією, що визначає ту функціональну частину залежної

UA 77187 C2

функції, що виконується у пристрої (6),

– або за допомогою використання поля (CII) ідентифікації інструкції і полів (CIP_k) ідентифікації, передбачених для операндів.

12. Спосіб за п. 9, 10 або 11, який **відрізняється** тим, що:

а у фазі (P) захисту визначають:

– як метод перейменування установних параметрів – метод кодування для кодування установних параметрів,

– і як засоби (20) відновлення – засоби, що застосовують метод декодування для розкодування перейменованих установних параметрів і відновлення ідентичності залежних функцій, що їх потрібно виконати у пристрої (6).

13. Спосіб за будь-яким з пп. 1-12, який **відрізняється** тим, що:

а у фазі (P) захисту:

• модифікують захищену програму (2p):

– за допомогою вибору у вихідному коді (2ps) захищеної програми щонайменше одного умовного переходу, виконуваного щонайменше в одному вибраному алгоритмі,

– за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми, причому зазначена модифікація така, що в ході виконання захищеної програми (2p) функціональна можливість щонайменше одного вибраного умовного переходу виконується, за допомогою другої виконуваної частини (2peu), у пристрої (6),

– і за допомогою створення:

Ш першої частини (2pos) об'єктного коду захищеної програми (2p), причому перша частина (2pos) об'єктного коду така, що в ході виконання захищеної програми (2p) функціональна можливість щонайменше одного вибраного умовного переходу виконується у пристрої (6),

Ш і другої частини (2pou) об'єктного коду захищеної програми (2p), причому друга частина (2pou) об'єктного коду така, що після завантаження у пристрій (6), у ході виконання захищеної програми (2p) реалізується друга виконувана частина (2peu), за допомогою якої виконується функціональна можливість щонайменше одного вибраного умовного переходу,

а а у фазі (U) використання:

• за наявності пристрою (6), кожного разу, коли цього вимагає фрагмент першої виконуваної частини (2pes), виконують у пристрої (6) функції щонайменше одного вибраного умовного переходу таким чином, щоб указаний фрагмент виконувався коректно й, отже, захищена програма (2p) була повнофункціональною;

• а за відсутності пристрою (6), незважаючи на запит фрагмента першої виконуваної частини (2pes) на виконання функцій умовного переходу у пристрої (6), не забезпечується можливість коректної відповіді на вказаний запит, так що принаймні вказаний фрагмент не виконується коректно й, отже, захищена програма (2p) не є повнофункціональною.

14. Спосіб за п. 13, який **відрізняється** тим, що у фазі (P) захисту модифікують захищену програму (2p):

– за допомогою вибору у вихідному коді (2ps) захищеної програми щонайменше однієї серії вибраних умовних переходів,

– за допомогою модифікації щонайменше одного вибраного фрагмента коду (2ps) захищеної програми, причому згадана модифікація така, що в ході виконання захищеної програми (2p) глобальна функціональна можливість щонайменше однієї вибраної серії умовних переходів виконується у пристрої (6) за допомогою другої виконуваної частини (2peu),

– і за допомогою створення:

Ш першої частини (2pos) об'єктного коду захищеної програми (2p), причому перша частина (2pos) об'єктного коду така, що в ході виконання захищеної програми (2p) функціональна можливість щонайменше однієї вибраної серії умовних переходів виконується у пристрої (6),

Ш і другої частини (2pou) об'єктного коду захищеної програми (2p), причому друга частина (2pou) об'єктного коду така, що після завантаження у пристрій (6), у ході виконання захищеної програми (2p) реалізується друга виконувана частина (2peu), за допомогою якої виконується глобальна функціональна можливість щонайменше однієї вибраної серії умовних переходів.

15. Спосіб за будь-яким з пп. 1-14, який **відрізняється** тим, що розкладають фазу (P) захисту на вхідну підфазу (P_1) захисту, не залежну від програми, що захищається, і на вихідну підфазу (P_2) захисту, залежну від програми, що захищається.

16. Спосіб за п. 15, який **відрізняється** тим, що в ході вхідної підфази (P_1) захисту задіюють стадію (S_{11}) визначень, на якій виконуються всі визначення.

17. Спосіб за п. 16, який **відрізняється** тим, що після стадії (S_{11}) визначень задіюють стадію конструювання (S_{12}), на якій створюються засоби експлуатації.

18. Спосіб за п. 17, який **відрізняється** тим, що після стадії (S_{12}) конструювання задіюють стадію (S_{13}) передперсоналізації, яка полягає в тому, що завантажують у незадіяний пристрій (60) щонайменше частину засобів експлуатації, щоб одержати передперсоналізований пристрій (66).

19. Спосіб за п. 16 або 17, який **відрізняється** тим, що в ході вхідної підфази (P_1) захисту задіюють стадію (S_{14}) виготовлення засобів, на якій виготовляються засоби, що допомагають при створенні захищених програм або автоматизації захисту програм.

20. Спосіб за п. 15 або 18, який **відрізняється** тим, що вихідну підфазу (P_2) захисту розкладають на:

• стадію (S_{21}) створення, на якій на основі уразливої програми (2v) створюється захищена програма (2p);

• можливо, стадію (S_{22}) модифікації, на якій модифікується захищена програма (2p);

UA 77187 C2

• і, можливо, стадію (S_{23}) персоналізації, на якій:

– друга частина (2pou) об'єктного коду захищеної програми (2p), що, можливо, містить засоби експлуатації, завантажується щонайменше в один незадіяний пристрій (60), щоб одержати принаймні один пристрій (6),

– або друга частина (2pou) об'єктного коду захищеної програми (2p), що, можливо, містить засоби експлуатації, завантажується щонайменше в один передперсоналізований пристрій (66), щоб одержати щонайменше один пристрій (6).

21. Спосіб за п. 19 або 20, який **відрізняється** тим, що в ході стадії (S_{21}) створення і, можливо, стадії (S_{22}) модифікації використовують щонайменше один із засобів, призначених для використання при створенні захищених програм або автоматизації захисту програм.

UA 77187 C2

UA 77187 C2

Фіг. 11

Фіг. 12

Фіг. 13

UA 77187 C2

Фіг. 21

Фіг. 22

Фіг. 30

Фіг. 31

UA 77187 C2

Фіг. 40

Фіг. 41

Фіг. 42

UA 77187 C2

ФІГ. 69

ФІГ. 70

ФІГ. 71

UA 77187 C2

Фіг. 72

Фіг. 73

Фіг. 74

Фіг. 80

UA 77187 C2

UA 77187 C2

Фіг. 84

Фіг. 85

UA 77187 C2

Фіг. 90

Фіг. 91

Фіг. 92

UA 77187 C2

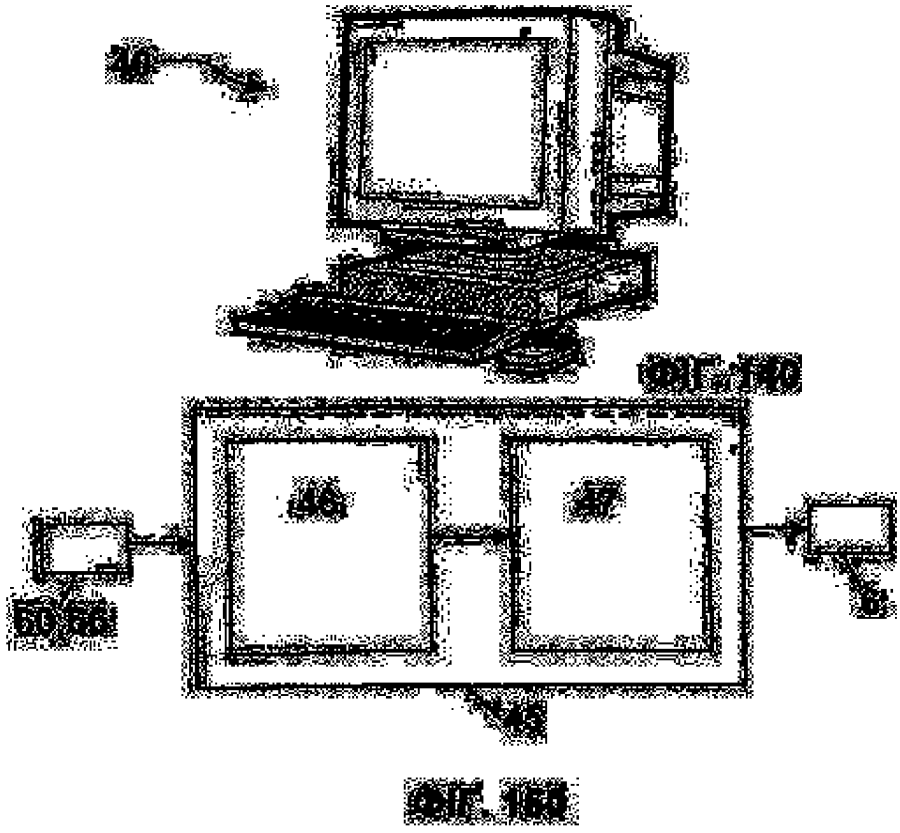
ФІГ.10а

ФІГ.10б

ФІГ.12б

ФІГ.13б

UA 77187 C2



ФІГ. 16Б

Офіційний бюлетень "Промислоава власність". Книга 1 "Винаходи, корисні моделі, топографії інтегральних мікросхем", 2006, N 11, 15.11.2006. Державний департамент інтелектуальної власності Міністерства освіти і науки України.

UA 77187 C2

UA 77187 C2