УДК 546.262
В.А.Васильків
Луцький національний технічний університет

ВИЗНАЧЕННЯ РІВНЯ ЗАХИЩЕНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МІКРОКОНТРОЛЕРІВ ЗА МЕТОДОМ АНАЛІЗУ СТРУМУ ЖИВЛЕННЯ

У даній статті проведено огляд методів дослідження ступеня захищеності программного забезпечення мікроконтролерів. Показано, що аналіз струму споживання мікроконтролера містить інформацію про виконувану мікроконтролером програми, у статті розглянуто визначеняя рівня захищеності за методом аналізу струму споживання.

Ключові слова: *мікроконтролер, програмне забезпечення, деструктивний метод, детекту-вання*.

Вступ

Недостатньо уваги приділяється правильному вибору мікроконтролерів для пристроїв, які призначення для захищених систем. В основному це трапляється через відсутність інформації про дійсний рівень безпеки мікропроцесорів. Виробники мікропроцесорів та мікроконтролерів не приділяють достатньо уваги створенню та тестуванню захисних механізмів у своїх пристроях. Навіть тоді, коли виробник заявляє що його пристрій обладнаний захистом від несанкціонованого втручання, він не гарантує, та не несе відповідальності за можливий успішний взлом мікроконтролерного пристрою. В цій ситуації, розробнику мікропроцесорної системи важливо знати реальну інформацію про ступінь захищеності використовуваних компонентів та знати надійний метод їх перевірки на надійність.

Методи дослідження ступеня захисту програмного забезпечення у мікропроцесорних системах

Оцінка захищеності програмного забезпечення мікроконтролера або мікропроцесора являє собою непросте завдання, оскільки така оцінка вимагає врахування багатьох факторів:

- тип упаковки (корпуса) мікросхеми;
- топологія кристала мікросхеми;
- тип та структура внутрішньої пам'яті;
- наявність інтерфейсів введення–виведення;
- алгоритми програмування та відладження.

Також варто врахувати наявність різноманітних механізмів захисту таких як, наприклад, Brown-out детектори, фільтри живлення, захисний екран, захисні Fuse-біти і т.д [1,2]. На сьогоднішній час не існує ніякого однозначного тесту чи методу, за яким можна визначити рівень захищеності інформації у мікроконтролері або у мікросхемі пам'яті. Можна лише спробувати виконати той чи інший тип атаки та визначити стійкість системи до даного типу атак. Варто застосувати різні методи атак та оцінювати успішність або неуспішність тієї чи іншої атаки. Чим більше різних типів атак проведено, тим більше інформації можна отримати про захищеність системи. Методи атак бувають деструктивними, тобто такими, що вимагають відкриття корпуса мікросхеми, та недеструктивними.

Деструктивні методи атак вимагають наявності прямого доступу до внутрішніх компонентів пристрою [1]. Необхідним обладнанням для даного типу атак є наявність спеціального оптичного або електронного мікроскопа з великою міжфокусною відстанню та рецизійного щупа, який дозволяє приєднувати зондуючий контакт до електричних провідників досліджуваного кристала, причому точність встановлення зонда повинна бути не більше мікрона [1]. Перевагою деструктивних методів є їх потенційна можливість зчитування інформації із внутрішньої пам'яті. Недоліком деструктивних методів є потреба у дорогому обладнанні, та необхідності вивчення структури кристалу, що є досить важкою процедурою, оскільки кількість транзисторів на кристалі у сучасних мікросхемах перевищує мільйони, при чому розташування транзисторів може бути багатошаровим, що затрудняє вивчення структури кристалу та пошук необхідних для модифікації точок.

Недеструктивні методи [1] дозволяють отримати деяку інформацію про внутрішній стан мікроконтролера без зняття упаковки мікросхеми. Розглянемо наступні недеструктивні методи: Дослідження часу виконання програми. Якщо відомо що програма у мікроконтролері оброблює великі масиви даних, то, подаючи вхідні масиви даних різного розміру та різного наповнення та вимірюючи час обробки цих даних, можна встановити алгоритм кодування або перетворення інформації.

Атаки з повним перебором. Якщо треба визначити пароль доступу до захищеної інформації, використовують повний перебір усіх можливих комбінацій паролів та ключів. Якщо додатково щось відомо про алгоритм перевірки пароля, можна значно скоротити час перебору. Ще одним варіантом методу атак з повним перебором є подавання на всі контакти контролера різноманітних комбінацій сигналів з метою входження в режим покрокового виконання або налагоджування. Аналіз струму споживання мікроконтролера. Під час переключення транзисторних структур у мікросхемах виникають перехідні процеси. Перехідні процеси зумовлені наявністю у МДН-транзисторах внутрішніх ємностей. При переключенні транзисторного інвертора завдяки ємності виникає короткочасне коротке замикання по шині живлення.

Такі короткі замикання приводять до сплесків у струмі споживання, які можна детектувати за допомогою чутливого прилада. З вищесказаного можна зробити висновок, що скачки струму залежать не стільки від стану певних модулів мікроконтролера, скільки від зміни цих станів. Вимірювання струму споживання дозволяє визначити, наприклад, кількість змінених бітів шини, і таким чином визначити інструкцію, чи оброблювані дані. Можливо також визначити тип інструкції, що виконується, встановивши, які арифметично-логічні чи иконавчі модулі вона задіює під час свого виконання.

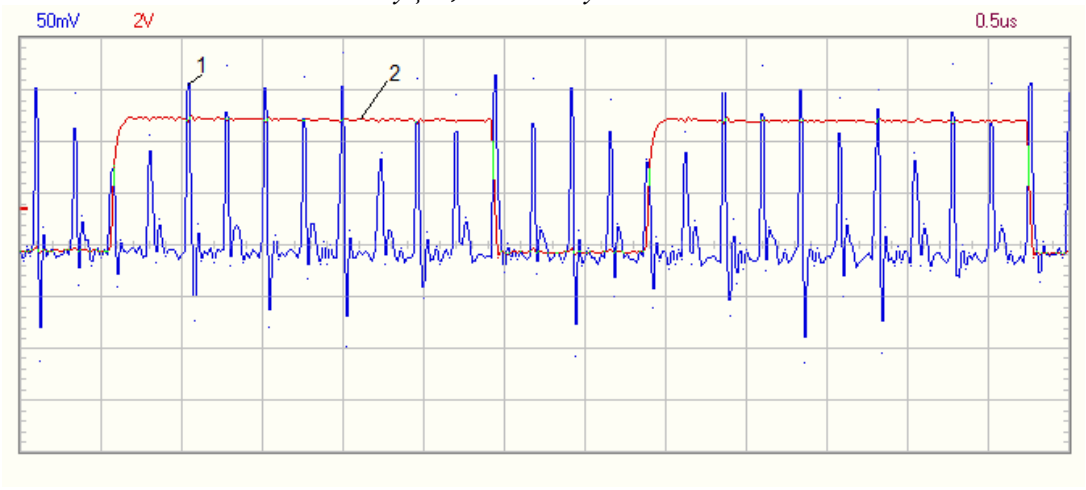
Достоїнством наведеного методу є відносно невисока вартість обладнання, оскільки для виконання дослідження необхідно мати лише осцилограф з широкою смугою пропускання та комп'ютер. Недоліком наведеного методу є відносно мала його інформативність. Тому перспективним шляхом удосконалення цього методу є застосування спектральних та вейвлет методів обробки сигналів, в чому і полягає мета даної статті.

Атаки з продукуванням збоїв. Під час роботи контролера змінюють певні параметри зовнішніх впливів, такі як: збільшення або зменшення робочої частоти контролера за межі штатного діапазону, збільшення або зменшення напруги живлення контролера, збільшення температури. Також можливе введення тимчасових провалів або завад до напруги живлення або тактових сигналів контролера. Результатом таких впливів може бути порушення виконання інструкцій контролера, що призводить до збою в роботі внутрішньої програми, та як результат, до викриття певних захищених даних або ключів шифрування

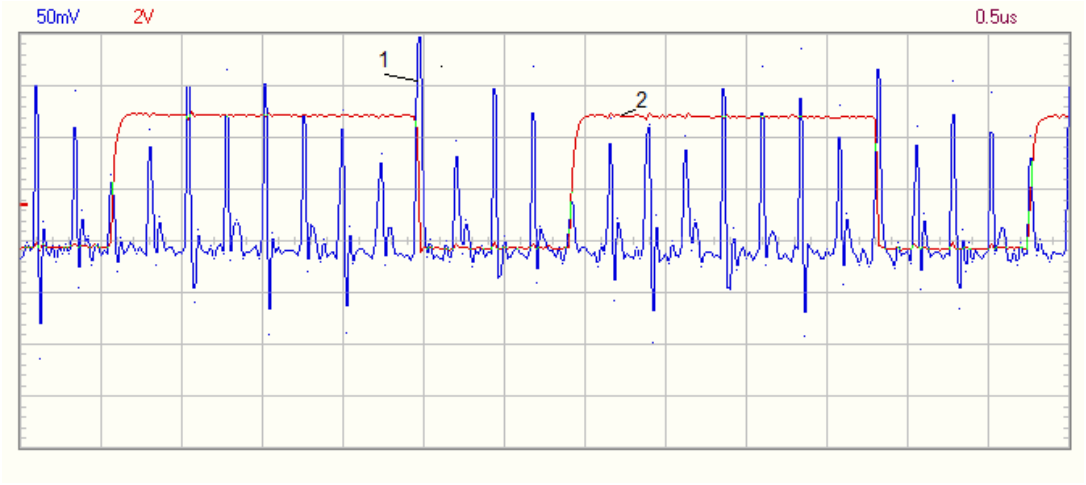
Дослідження струму споживання мікроконтролера

При проведенні експерименту були використані: плата мікроконтролера ATTiny12, датчик струму, цифровий осцилограф PCSU1000, програматор та персональний комп'ютер. Після завантаження чергового набору програм у пам'ять мікроконтролера виконувалося 3-х кратне знімання осцилограм струму для кожної програми. Кожна осцилограма зберігалася як у графічному вигляді, так і в текстовому. Осцилограми для команд мікроконтролера CP та BRCS у графічному вигляді наведено відповідно на Рис.1-а. та Рис.1-б. При цьому: 1 – корисний сигнал, 2 – сигнал синхронізації, також виробляється мікроконтролером.

Результати усіх вимірювань зберігалися у файлах даних програми Oscilloscope. Таким чином, створювалася база даних команд та відповідних їм струмів споживання. Отримана база даних використовується як джерело вихідних даних для подальших розрахунків.



а



б

Рис. 1. Приклад осцилограм струму для: а) – команди CP: та б) – команди BRCS

Рівень захищеності рограмм ого забезпечення мікроконтролера можна визначити як співвідношення кількості успішно виділених команд до загальної кількості осцилограм струмі, що оброблювались. Створення алгоритму ідентифікації усіх виконуваних команд утруднене внаслідок залежності струму споживання мікроконтролера не тільки від внутрішньої програми, але й від його архітектури, відхилення параметрів, наявності зовнішніх ланцюгів. Тому для зменшення можливого впливу вказаних параметрів, дослідження було зведено до одного екземпляру мікроконтролера та до спрощених рограмм. Крім того, буловилучено зовнішні ланцюги введення та виведення сигналів, для того, щоб вони не впливали на результати експериментів.

У якості критерію, що показує ступінь схожості струмів споживання між собою при обробці результатів вимірювання використовується коефіцієнт взаємної кореляції [4]. Розрахунок коефіцієнту взаємної кореляції виконується наступним чином (1):

$$\frac{\sum_{n=0}^{N-1} x_1(n)x_2(n)}{\sum_{n=0}^{N-1} x_1^2(n) * \sum_{n=0}^{N-1} x_2^2(n)} \quad (1)$$

де x1(n),x2(n) – послідовності, що порівнюються; n– кількість членів послідовності.

За формулою (1) можна обчислити кореляцію лише двох послідовностей однакової довжини. У випадку послідовностей неоднакової довжини треба враховувати вплив крайового ефекту [4]. Значення коефіцієнта взаємної кореляції лежить між -1 та +1, при чому значення +1 вказує на повне співпадіння послідовностей, а значення -1 вказує на те, що сигнали знаходяться в протифазі. Значення 0 вказує на те, що сигнали абсолютно відмінні, наприклад якщо один з сигналів є шумовим.

Порівнюючи за допомогою коефіцієнта взаємної кореляції струмів споживання між собою, можна зробити висновок про ступінь схожості еталонного значення струму та значення струму,

що вимірюється. Це дозволяє детектувати мікропроцесорні команди, значення струму яких вибирається із заздалегідь створеної бази даних.

Оцінка кореляції проводиться у часовій області, у області вейвлетів, та у частотній області за допомогою швидкого перетворення Фур'є. Необхідність дослідження кореляції у вказаних областях викликана необхідністю пошуку оптимального та ефективного алгоритму визначення мікропроцесорної команди по її струму споживання.

Програмне забезпечення у системі MatLab [3] для визначення коефіцієнта взаємної кореляції між вимірюваними струмами мікропроцесора виконує обчислення та обробку даних, формує таблиці кореляційних коефіцієнтів та оцінює коефіцієнти взаємної кореляції. В тому випадку, якщо коефіцієнт кореляції при порівнянні менший 0,9 програма робить висновок про те, в якій області можуть бути розрізнені дані дві команди: у часовій області, у частотній області, або після розкладання за допомогою вейвлетів Хаара [5,6] та ОБ [5] та порівняння відповідних коефіцієнтів деталізації.

Програма підраховує кількість успішних порівнянь двох струмів споживання, в результаті яких ці струми виявляються різними, та обчислює ефективність кожного методу порівняння за наступною формулою:

$$r = \frac{n_i}{N} * 100\% \qquad (2)$$

де N – загальна кількість порівнянь команд; ni - кількість успішних порівнянь за окремим методо. У результаті одержуємо наступні показники ефективності використаних методів (рис. 2):

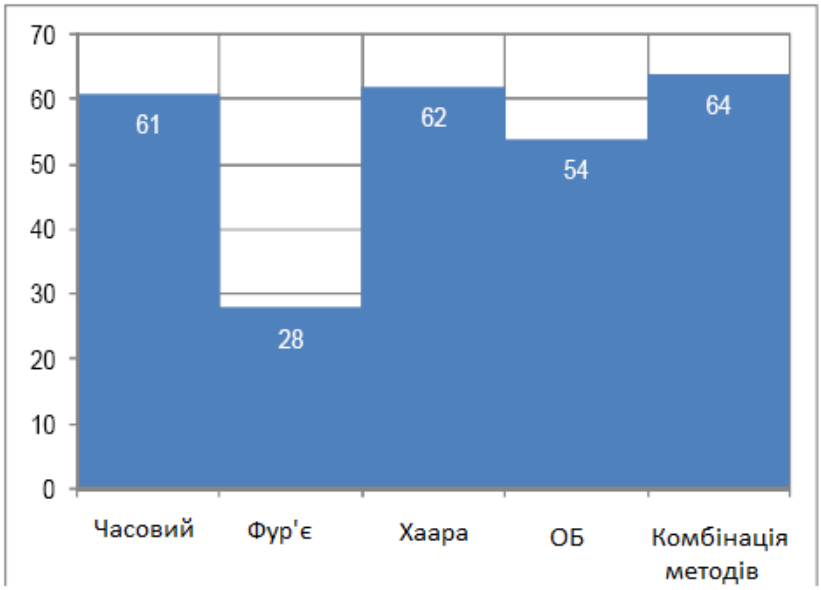


Рис. 2. Ефективність методів детектування, %

Очевидно, що для виділення більшості команд у 61% випадків достатньо використання порівняння

струмів споживання у часовій області. Дещо підвищити ефективність даного методу можна за умови використання вейвлет-перетворень Хаара та ОБ (відповідно 62% та 54%). Низькою ефективністю у порівнянні з іншими методами володіє Фур'є перетворення (28%), оскільки воно дає менше інформації про миттєві значення сигналу, і призначене для дослідження періодичних сигналів. У 29% випадків порівняння струмів різних команд не виявило між ними відмінностей. Загальна кількість команд, яка може бути виділена комбінацією наведених методів складає 64%.

Незважаючи на те, що метод порівняння у часовій області є досить ефективним, у деяких випадках, команди відокремити можна лише після проведення порівняння у області вейвлетів, це стосується зокрема виділення команд BRMI, BPRPL, CPSE, COM, NEG та деяких інших.

Висновки

1) Найбільшою ефективністю виділення комнад характеризується порівняння у області вейвлетів Хаара та ОБ (ефективність відповідно 62% та 54%), оскільки порівнюється інформація про флуктуації сигналів. Найнижчою ефективністю характеризується порівняння у частотній

області за допомогою перетворення Фур'є (ефективність 28%). У 29% випадків порівняння струмів різних команд не виявило між ними відмінностей. Загальна кількість команд, яка може бути виділена комбінацією наведених методів складає 64%.

2) За допомогою методу дослідження струму споживання мікроконтролера, у загальному випадку неможливо отримати вихідний асемблерний код програми. Однак можна зробити висновки про те, які алгоритми в даний момент виконує мікроконтролер, наприклад алгоритми обчислення, вводу-виводу даних, програмування та ін.

3) Наведений мікроконтролер має недостатній рівень захищеності, що виявляється в можливості детектування значної кількості команд.

Література

1. Жуйков В.Я., Терещенко Т.А., Петергеря Ю.С. Спектральные преобразования функций cтичным аргументом: теория и применения. – К.: Аверс, 2006. – 293 с

2. Преобразование дискретных сигналов на конечных интервалах в ориентированом базисе / Жуйков В.Я., Терещенко Т.А., Петергеря Ю.С. – К.: Аверс, 2004. – 274 с.

3. Ануфриев И.Е. Самоучитель MatLab 5.3/6.x. СПб.: БХВ-Петербург, 2004. – 736 с.: ил.

4. Айфичер Эммануил С., Джервис Барри У. Цифровая обработка сигналов: Практический подход, 2-е издание. : Пер. с англ – М.: Издательский дом «Вильямс», 2004. – 992 с. : ил.

5.. Semi-invasive attacks – A new approach to hardware security analysis. Sergei P. Skorobogatov, University of Cambridge, April 2005. – 144 p.

6. Евстифеев А.В. Микроконтроллеры AVR семейства Tiny и Mega фирмы "Atmel" М.: Издательский дом «Додэка-XXI», 2004 – 560 с.