УДК 638.235.231

В.А. ГОРБАЧЕВ, канд. техн. наук, проф., К.Б. АБДУЛРАХМАН

ОБЗОР ПРОБЛЕМ БЕЗОПАСНОСТИ И ПРОЕКТИРОВАНИЯ ЗАЩИЩЕННЫХ ЭЛЕКТРОННЫХ СИСТЕМ

1. Введение. Актуальность

До некоторого времени безопасность компьютерной системы традиционно связывали с безопасностью программного обеспечения или обрабатываемой информации. Аппаратные ресурсы, используемые для обработки информации, считались надежными. Появление аппаратных закладок (АЗ), в зарубежной литературе известных как Hardware Trojan (HT), и угроз, связанных с ними, нарушило это доверие. АЗ могут быть реализованы в ASIC и в FPGA, в имеющихся на рынке микропроцессорах, микроконтроллерах, сетевых процессорах или цифровых процессорах сигналов (DSP). Таким образом, требование безопасности электронных систем – это такое же ограничение, как низкая потребляемая мощность, высокая скорость, устойчивость к отказам и т.д.

Приведем несколько важных характеристик АЗ:

1. Сложность ИС и чрезвычайно малые размеры АЗ делают практически невозможным ее обнаружение без специальных инструментальных средств.

2. Даже в случаях, когда факт нарушения безопасности будет выявлен, доказать, что это действие выполнено АЗ, очень сложно.

3. АЗ обладают свойством перманентности: как только в систему была встроена АЗ, угроза сохраняется всегда, когда система находится во включенном состоянии.

4. АЗ расположены ниже программного блока, включающего операционную систему, программное обеспечение (middleware), работающее поверх операционной системы, и приложения. Это позволяет АЗ или полностью обойти традиционные программные средства защиты информации, или сделать их малоэффективными.

Эти и другие характеристики делают такие закладные устройства очень перспективным элементом при планировании электронных диверсий.

2. Классификация АЗ

Разработка эффективных методов, предназначенных для обнаружения и блокировки АЗ зависит от полноты характеристик АЗ и схемы классификации.

Классификация АЗ – это дерево, где каждая ветвь определяет другую характеристику (или атрибут) АЗ. В идеальном случае определенная АЗ должна находиться только на одном листе дерева.

Классификация АЗ позволяет определить значимость каждой из характеристик АЗ, ее влияние на систему, а также выявить этапы разработки системы, на которых АЗ могут быть устранены. Кроме того, классификация АЗ имеет и самостоятельное значение – ее можно использовать как основу для анализа и разработки методов обнаружения и предупреждения АЗ данных классов. Более того, классификация АЗ позволяет осуществить экспериментальное тестирование систем защиты, т.к. она позволяет выбрать направления и способы проведения тестовых атак на наиболее уязвимые компоненты систем защиты.

Существует несколько методов классификации АЗ. Эти методы используют различные характеристики АЗ. Наиболее распространенными являются следующие характеристики АЗ: физические свойства, механизм активации и функциональное действие.

Например, в [1, 2] АЗ классифицируются по трем признакам: физическим свойствам, механизмам активации и функциональному воздействию. Подробная классификация АЗ, где рассмотрены многие характеристики, рассмотрена в [3, 4] и представлена на рис. 1.

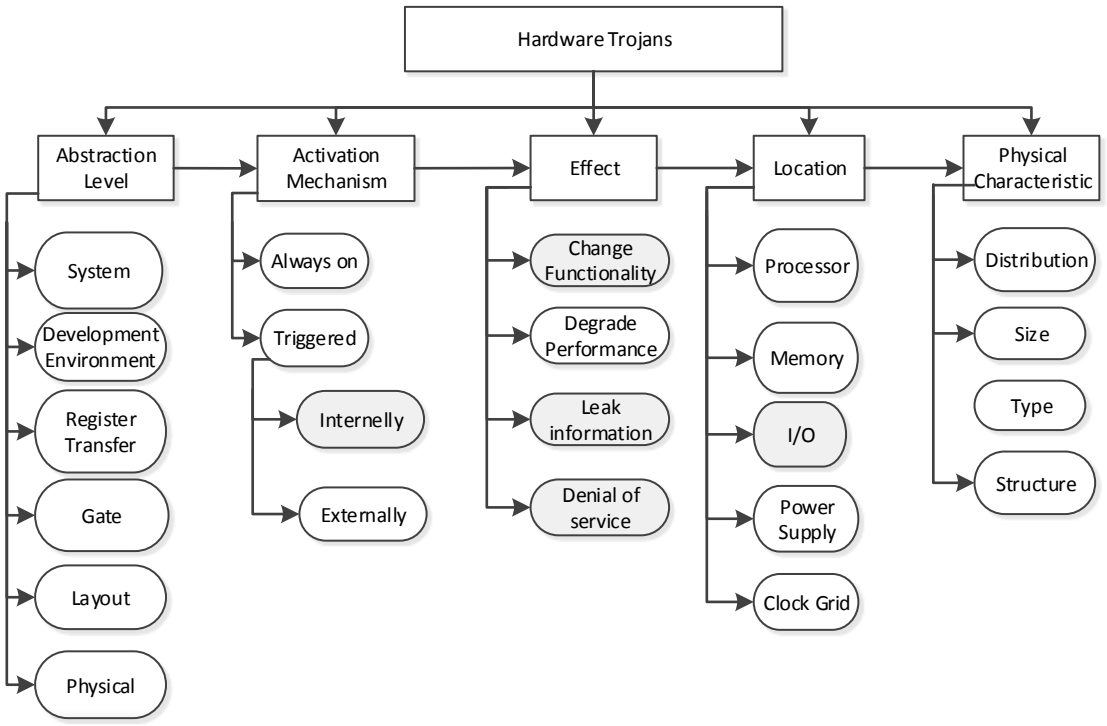

Рис. 1

Учитывая то, что угрозы АЗ направлены на нарушение безопасности информации, в работе [5] предлагается следующая классификация АЗ по их функциональному воздействию (Effect): нарушающие конфиденциальность, целостность и доступность информации.

Важно отметить, что сложность АЗ изменялась от простых, встроенных, например, в клавиатуру [6], до сложных, встроенных, например, в процессор [7].

В результате анализа литературных источников по классификации АЗ сделаем следующие выводы:

1. Множество угроз постоянно расширяется и имеет тенденцию экспоненциального роста. Это означает, что невозможно создать исчерпывающую классификацию АЗ [4].

2. Пространство проектных решений АЗ слабо изучено. Ошибочно утверждать, что целью АЗ является атака на аппаратные ресурсы системы. Возможность атаки АЗ на программный стек, лежащий выше аппаратных ресурсов, является реальной.

3. Электронные системы могут быть заражены одновременно несколькими АЗ, которые могут совместно разрушить систему защиты.

4. Структура АЗ может быть распределенной, что значительно увеличивает сложность ее обнаружения.

3. Методы борьбы с АЗ

Существуют два основных способа [8], обеспечивающих гарантии того, что, используемая ИС является аутентичной, другими словами, она выполняет только те функции, которые были определены первоначально, и не более того. Первый способ – сделать весь процесс разработки ИС надежным. Этот способ чрезмерно дорогостоящий и практически невозможный, с учетом текущих тенденций в глобальном распределении процессов проектирования и изготовления ИС. Второй способ – проверить аутентичность уже готовых ИС перед их использованием.

Рассмотрим классификацию методов борьбы с АЗ, представленную на рис. 2 [4]. Эти методы можно разделить на два класса: методы обнаружения АЗ и методы предотвращения внедрения АЗ.

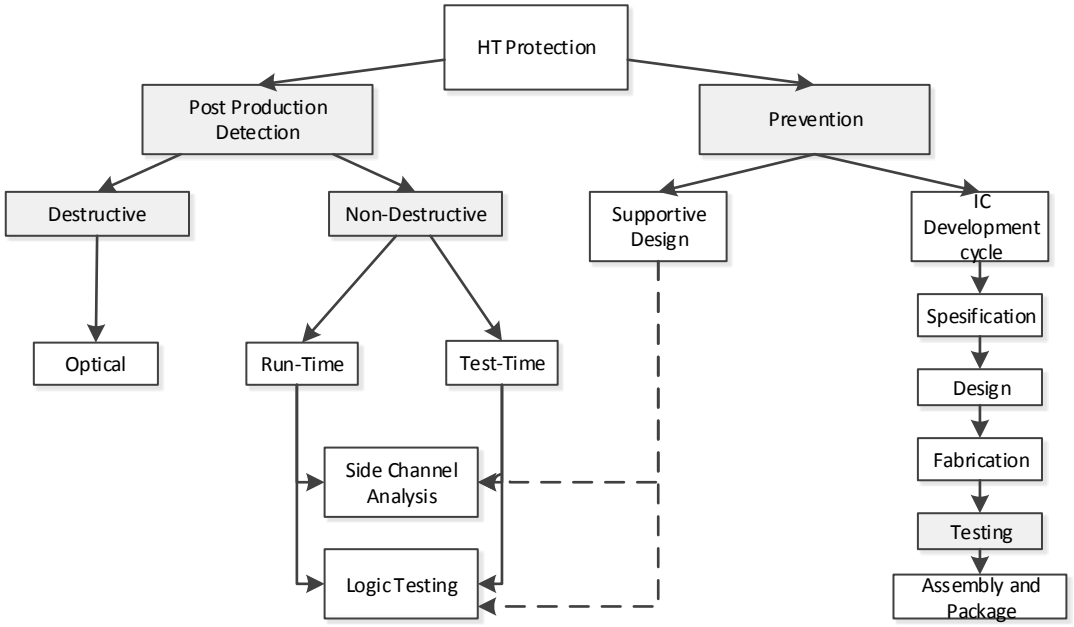

Рис. 2

3.1. Методы обнаружения угроз АЗ

Известно, что невозможно полностью предотвратить внедрение АЗ в ИС на этапе ее производства [8]. В тех случаях, когда превентивные меры, использующиеся для защиты от АЗ, не приносят результата, используются методы обнаружения АЗ, внедренных в структуру ИС. Методы обнаружения АЗ применяются после производства ИС. Существует множество различных методов обнаружения АЗ. Кратко рассмотрим классификацию этих методов (рис. 2), приведенную в [2, 4]. Методы обнаружения АЗ можно представить в виде двух классов: методы, разрушающие и неразрушающие ИС.

Методы, разрушающие ИС. Применяются для определения полной электрической и функциональной схемы. В основе метода лежит химическая полировка металлов, а затем использование сканирующего электронного микроскопа. Полностью уничтожает исследуемую ИС. Характеризуется высокой трудоемкостью и большими материальными затратами.

Методы, не разрушающие ИС. Неразрушающие методы обнаружения АЗ не разрушают ИС и классифицируются как методы, которые обнаруживают АЗ либо во время функционирования (Run-Time), либо во время логического тестирования (Test-Time) интегральной схемы.

Run-Time методы [13] *обнаружения АЗ.* В настоящее время существует большое разнообразие методов обнаружения АЗ, которые осуществляют непрерывный мониторинг характеристик ИС в реальном времени.

Например, авторы [9] подробно описывает метод обнаружения двух типов атак на память ИС: DoS-атака и атака, при которой АЗ отключает защиту памяти. Для обнаружения и блокировки атак предлагается использовать дополнительный защитный модуль, осуществляющий контроль операций обращения к памяти. Этот метод требует, чтобы операционная система была изменена с учетом взаимодействия с защитным модулем.

В [10] для мониторинга безопасности функционирования ИС в режиме реального времени авторы добавили реконфигурируемую логику DEsign-For-ENabling-SEcurity (DEFENSE). После изготовления микросхемы реконфигурируемая логика программируется, подробно описывая поведение ИС. Отклонения в поведении ИС могут быть обнаружены в процессе ее функционирования.

Авторы [11] предлагают обнаруживать АЗ, выполняя функционально эквивалентные процессы на нескольких аппаратных компонентах обработки информации. Затем результаты можно сравнить, выявляя процессы, на которые повлияла АЗ.

В результате анализа методов обнаружения АЗ, осуществляющих непрерывный мониторинг характеристик ИС в реальном времени, приходим к следующим выводам. Run-Time методы обнаружения АЗ:

1. Способны обнаружить только определенный тип АЗ.

2. Являются эффективными при условии, что АЗ находится в активном состоянии.

3. Позволяют осуществлять мониторинг характеристик ИС в реальном времени в критических режимах, в режиме ожидания, а также оценивать политику безопасности, производительность и доступность модулей системы.

Test-Time методы обнаружения АЗ. Test-Time методы обнаружения АЗ, в свою очередь, делятся на два класса: методы логического тестирования и методы, использующие анализ побочных каналов.

Методы логического тестирования (Logic Testing). Много работ посвящено развитию методов логического тестирования**.** Достаточно подробный анализ работ, посвященных данному подходу, можно найти в [2, 3, 6, 12, 13]. Авторы этих работ утверждают, что огромное логическое пространство состояний современной ИС делает невозможным, с вычислительной точки зрения, построение тестового вектора, покрывающего все логическое пространство ИС.

В результате анализа методов логического тестирования, приходим к выводам:

1. Тесты, используемые для обнаружения производственных ошибок, например таких как константная неисправность и задержки, не могут гарантировать обнаружение АЗ. Такие тесты работают со списком соединений ИС, свободной от АЗ, и поэтому не могут активировать и обнаруживать АЗ.

2. Методы логического тестирования не могут обнаружить АЗ, которые не производят воздействия на функциональный выход ИС.

3. Очень проблематично построить исчерпывающий тестовый вектор, обнаруживающий спусковые механизмы АЗ, срабатывающие от времени, например, такие, как time-bombs.

4. АЗ могут находиться в плохо контролируемых и доступных модулях, что делает маловероятным их активацию и обнаружение с использованием случайных или функциональных векторов.

Методы, использующие анализ побочных каналов (Side-Channel Analysis).

Метод анализа побочных каналов [13] использует тот факт, что сам механизм запуска и функционирование АЗ меняет определенные параметры ИС. К параметрам, свидетельствующим о наличии в структуре системы АЗ, относятся: изменение потребляемой мощности, задержки, токи утечки, повышение температуры определенной части ИС.

В [13] рассматривается факт изменения потребляемой мощности как параметр метода для обнаружения АЗ. Проведенные экспериментальные испытания подтвердили эффективность используемого анализа побочных каналов.

Авторы [14] демонстрируют эффективное использование метода побочных каналов, осуществляя сравнение энергопотребления между цепями, инфицированными АЗ, и теми, которые свободны от АЗ. Большие изменения в потреблении мощности могут свидетельствовать о постороннем оборудовании.

В работе [15], в качестве параметра метода анализа побочных каналов, используются задержки, вызванные цепями АЗ.

В результате анализа методов, *использующих анализ побочных каналов*, приходим к следующим выводам:

1. Требуется наличие подлинной, т.е. свободной от АЗ, ИС, которая должна использоваться для сравнения с тестируемой. В то же время, нет никакой гарантии, что оставшиеся ИС свободны от АЗ.

2. Учитывая большое количество IP-модулей, используемых в ИС, а также высокую сложность современных IP-модулей, выявление небольших вредоносных изменений является чрезвычайно сложным.

3. АЗ могут находиться в плохо контролируемых и доступных модулях

4. Совершенствование технологий литографии приводит к тому, что изменения, обусловленные АЗ, все меньше влияют на электрические параметров ИС. Таким образом, обнаружение АЗ с использованием простого анализа параметров сигналов будет неэффективным.

3.2. Методы предотвращения внедрения АЗ

В отличие от методов обнаружения, методы предотвращения угроз АЗ объединяют методы, которые препятствуют внедрению АЗ. Одним из способов гарантирования, что в ИС не будет внедрена АЗ, является жесткое управление жизненным циклом разработки ИС (IC development cycle) на всех его этапах (рис. 2). Это важное звено в стратегии эффективной защиты. Этапами жизненного цикла разработки ИС являются этапы: составления спецификации (Specification), проектирования (Design), изготовления (Fabrication), тестирования (Testing) и сборки (Assembly and Package). Авторы [2] утверждают, что только этапы спецификации и тестирования могут быть не уязвимыми внедрением АЗ. Все другие этапы на практике уязвимы из-за зависимости от сторонних поставщиков IP модулей, от инструментов проектирования и от процесса проектирования и производства.

На этапе проектирования АЗ могут быть внедрены кем-то из разработчиков, или включением в проект инфицированных IP модулей. Различные рекомендации, предотвращающие внедрение АЗ при разработке ИС, рассматриваются в [2, 8, 16].

Особую группу составляют методы (*Supportive design*), которые на этапе проектирования ИС встраивают дополнительные защитные механизмы, блокирующие функционирование предполагаемой АЗ. Эти механизмы могут повышать эффективность методов логического тестирования (Logic Testing) и методов, которые используют анализ побочных каналов (Side-Channel Analysis).

Так, в [17] авторы рассматривают различные защитные механизмы и их практическую реализацию, которые управляют доступом и использованием данных в системе. Эти механизмы блокируют функционирование АЗ определенного типа.

Различные решения по встраиваемым дополнительным защитным механизмам, которые блокируют функционирование определенного типа АЗ, можно найти в [18, 19].

В результате анализа методов предотвращения угроз, приходим к выводу:

- эти методы характеризуются высокими накладными затратами;
- ориентированы на определенные типы АЗ;
- очень сложно полностью предотвратить внедрение АЗ в ИС на этапе ее производства.

4. Выводы. Перспективные направления исследований

Подведем некоторые итоги и сформулируем направление будущих исследований .

Во-первых, современные методы обнаружения, а также методы предотвращения внедрения АЗ не могут обеспечить полную гарантию того, что ИС или электронная система свободны от АЗ.

Во-вторых, рассматриваемые методы способны обнаруживать только определенный тип АЗ.

В-третьих, средства осуществления угроз безопасности (АЗ) выбираются не случайным образом. Новая эффективная АЗ непременно использует определенные особенности архитектуры и функционирования или недостатки средств защиты электронной системы.

В-четвертых, противостояние АЗ и средств защиты напоминает систему с обратной связью – новые типы АЗ приводят к появлению новых средств защиты, а недостатки в средствах защиты приводят к появлению новых типов АЗ и т.д. Разорвать эту обратную связь бесконечного противостояния можно двумя путями:

- создать эффективные и безупречно надежные средства защиты от каждого типа АЗ, или

- устранить причины указанных недостатков электронных систем, которые служат источником успешной реализации угроз безопасности.

Рассмотрим недостатки и преимущества того и другого метода.

Создание средств защиты от каждого вида угроз. К преимуществам данного метода следует отнести то, что средства защиты не зависят напрямую от назначения электронной системы и не требуют модификации по мере ее развития. Недостатки такого подхода очевидны: для создания эффективного механизма защиты необходимо проанализировать все типы АЗ и разработать для каждого типа соответствующий механизм противодействия. Практика показывает, что данный путь трудно осуществим вследствие следующих факторов:

- множество АЗ постоянно расширяется и имеет тенденцию экспоненциального роста. Это означает, что все время будут появляться новые АЗ, требующие новых мер защиты, т. к. старые против них бессильны;

- множество АЗ растет не только количественно, но и качественно, т. к. для того, чтобы АЗ состоялась, она должна принципиально отличаться от тех, на которые рассчитаны системы защиты. Это означает, что невозможно создать исчерпывающую классификацию угроз безопасности и предсказать появление новых типов АЗ.

Устранение причин, обуславливающих успешную реализацию угроз. Этот подход основан на проектировании защищенных систем обработки информации, устраняет причины появления изъянов защиты. Он должен удовлетворять следующим требованиям:

1. Интеграция средств защиты информации, в качестве обязательного элемента, в процесс ее обработки.

2. Включение в модель безопасности (или в механизмы защиты) функций, обеспечивающих безопасность электронной системы в условиях возможного появления внутри ее компонентов, осуществляющих деструктивные действия.

3. Используемые методы проектирования должны основываться на формальных принципах, доказательно обеспечивающих гарантии защищенности систем.

Преимущества этого подхода очевидны: он не зависит от развития АЗ, так как ликвидирует причину, а не следствие, поэтому он более эффективен, чем создание средств защиты от каждого вида АЗ. В качестве недостатков данного подхода можно отметить необходимость применения новых технологий и формальных методов проектирования защищенных электронных систем.

Целью последующего исследования являются *технологии проектирования защищенных ЭС,* обеспечивающие устойчивость ЭС к деструктивному воздействию внутренних компонентов.

Список литературы: 1. *Tehranipoor, M. and Koushanfar, F.* A Survey of Hardware Trojan Taxonomy and Detection // IEEE Design & Test of Computers, vol. 27, no. 1, pp. 10-25, Jan. 2010. 2. *Chakraborty, R. S., Narasimhan, S. and Bhunia, S.* Hardware Trojan: Threats and emerging solutions // IEEE International High Level Design Validation and Test Workshop. IEEE, Nov. 2009, pp. 166-171. 3. *He Li, Qiang Liu, Jiliang Zhang* Survey of hardware Trojan threat and defense // INTEGRATION, the VLSI journal 55, 2016, pp. 426–437. 4. *Julien Francq, Hardware Trojans* Detection Methods, Cassidian Cybersecurity // All rights reserved, in TRUDEVICE, 2013, Page 36 – 40. 5. *Gorbachov, V.* Malicious Hardware: characteristics, classification and formal models // IEEE East-West Design & Test Symposium (EWDTS2014), Kiev, Ukraine, 2014, pp. 254-257. 6. *Горбачев В.А., Степаненко, В.В.* Сертификация периферийных устройств компьютерных систем // Радиотехника. – 2003. – Вып. 134.- С. 206-209. 7. *Samuel, T. King and all* Designing and implementing malicious hardware // LEET'08 Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats Article No. 5 San Francisco, California – April 15 – 15, 2008. 8. *Mark Beaumont, Bradley Hopkins and Tristan Newby*. Hardware Trojans – Prevention, Detection, Countermeasures (A literature Review). DSTO Defence Science and Technology Organisation Edinburgh, South Australia 5111, Australia 2011]. 9. *Bloom, G., Narahari B. & Simha, R.* (2009) OS support for detecting trojan circuit attacks, in IEEE International Symposium on Hardware-Oriented Security and Trust. 10. *Abramovici, M. & Bradley, P.* (2009) Integrated circuit security: new threats and solutions, in Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, CSIIRW '09, ACM, New York, NY, USA, pp. 55:1–55:3. 11. *McIntyre, D. R.,Wolff, F. G., Papachristou, C. A. & Bhunia S.* (2009) Dynamic evaluation of hardware

trust // IEEE International Symposium on Hardware-Oriented Security and Trust, pp. 108–111. 12. *Syed Kamran Haider, Chenglu Jin, Masab Ahmad, Devu Manikantan Shila, Omer Khan and Marten van Dijk*. Advancing the State-of-the-Art // Hardware Trojans Detection, University of Connecticut, 2016. 13. *Tehranipoor, M., Wang, C*. Introduction to hardware security and trust. – New York, Springer, 2011. 14. *Banga, M.* & Hsiao, M. S. (2009) A novel sustained vector technique for the detection of hardware Trojans // in VLSI Design 2009: Improving Productivity through Higher Abstraction, The 22nd International Conference on VLSI Design, New Delhi, India, 5-9 January 2009, IEEE, pp. 327–332. 15. *Jin, Y. & Makris, Y.* (2008) Hardware Trojan detection using path delay fingerprint, in IEEE International Symposium on Hardware-Oriented Security and Trust. 16. *Potkonjak, M.* (2010) Synthesis of trustable ics using untrusted cad tools // Proceedings of the 47th Design Automation Conference, pp. 633–634. 17. *Waksman, A. & Sethumadhavan, S.* (2011) Silencing hardware backdoors // Proceedings of the 32nd IEEE Symposium on Security and Privacy, May 2011. 18. *Hicks, M., Finnicum, M., King, S. T., Martin, M. M. K. & Smith, J. M. (*2010) Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically, Security and Privacy // IEEE Symposium on 0, 159–172. 19. *Silva, M. L. & Ferreira, J. C.* (2010) Creation of partial FP configurations at run-time // Proceedings of the 2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, DSD '10, IEEE Computer Society, Washington, DC, USA, pp. 80–87,

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 07.09.2017