

Обзорная статья

DOI 10.66424/2071-8217-2026-3-12

УДК 004.056

СИСТЕМАТИЗАЦИЯ КАНАЛОВ И СРЕДСТВ ПРОВЕДЕНИЯ АТАК НА ВСТРАИВАЕМЫЕ СИСТЕМЫ ПОСРЕДСТВОМ ВНЕДРЕНИЯ ПОМЕХ

Г. Ю. Пагуба*, Д. Н. Кинешев, Е. Д. Кондачков

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *paguba_gyu@spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Пагуба Г. Ю., Кинешев Д. Н.,
Кондачков Е. Д. Систематизация
каналов и средств проведения
атак на встраиваемые системы
посредством внедрения помех // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 162–178. DOI: 10.66424/2071-8217-2026-3-12

ПОСТУПИЛА 24.07.2026

ПРИНЯТА 27.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Пагуба Г. Ю., Кинешев Д. Н.,
Кондачков Е. Д.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Выполнена систематизация каналов и средств проведения атак на встраиваемые системы посредством внедрения помех. Для каждого канала и средства рассмотрены его достоинства и недостатки, а также представлено его краткое описание. На основании проведенного анализа сформулированы направления дальнейших исследований, связанные с обеспечением устойчивости программного обеспечения к атакам посредством внедрения помех.

КЛЮЧЕВЫЕ СЛОВА

Информационная безопасность, поиск ошибок, программное обеспечение, внедрение помех, встраиваемые системы

Review article

DOI 10.66424/2071-8217-2026-3-12

SYSTEMATIZATION OF CHANNELS AND TOOLS FOR CONDUCTION OF ATTACKS ON EMBEDDED SYSTEMS VIA FAULT INJECTION

G. Yu. Paguba*, D. N. Kineshev, E. D. Kondachkov

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *paguba_gyu@spbstu.ru

FOR CITATION

Paguba G. Yu., Kineshev D. N.,
Kondachkov E. D. Systematization of
channels and tools for conduction

ABSTRACT

This paper systematizes channels and tools for conduction of attacks on embedded systems via fault injection. Each channel and tool's advantages and disadvantages are discussed, along with a brief descrip-

of attacks on embedded systems via fault injection. *Problems of information security. Computer systems*. 2026. No. 3, pp. 162–178. DOI: 10.66424/2071-8217-2026-3-12 (In Russian)

RECEIVED 24.07.2026

ACCEPTED 27.08.2026

PUBLICATION 13.10.2026

tion. Based on this analysis, directions for further research related to ensuring software resilience to fault injection are identified.

KEYWORDS

Information security, errors detection, software, fault injection, embedded systems

1. ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ

Во всем мире наблюдается тенденция поставки и использования встраиваемых систем, поставляемых без исходных кодов и формального описания используемых в их составе комплектов программного и аппаратного обеспечения соответственно. Потребители таких систем вынуждены руководствоваться информацией о принципах их работы, представленной в документации, без возможности ее независимой верификации. Таким образом, отсутствие локализации производства и высокая степень закрытости компонентов встраиваемых систем приводят к тому, что используемые системы могут не соответствовать требованиям к отсутствию ошибок и недекларированных возможностей. Описанные возможности могут позволить злоумышленнику, получившему доступ к встраиваемой системе, реализовать последовательность действий, которая в дальнейшем приведет к различным нежелательным результатам, например, к получению полного доступа к устройству, внедрению в состав встраиваемой системы вредоносного программного обеспечения (ПО) и т.д. В частности, при отсутствии возможности эксплуатации уязвимостей в ПО злоумышленники могут применять подходы по манипулированию физическими свойствами атакуемых встраиваемых систем.

В источнике [1] продемонстрирована возможность повышения привилегий во встраиваемом ПО пользовательского терминала системы Starlink посредством кратковременного изменения напряжения на

линии питания данной системы. Повышение привилегий, как описано в источнике, достигалось за счет компрометации цепочки доверенной загрузки устройства. Во время проверки криптографической подписи и хеша загрузчика BL2 устройства, внедрялись помехи на линии питания основного микропроцессора, что привело к успеху в проверке вне зависимости от достоверности образа проверяемого загрузчика. Таким образом стала возможной загрузка произвольных образов загрузчика BL2 и ядра операционной системы устройства и соответственно повышение привилегий атакующего до суперпользователя.

В источнике [2] продемонстрирована возможность проведения аналогичной атаки на мобильные устройства с процессорами Mediatek MT8163V. Как и в предыдущем случае, атака велась на цепочку доверенной загрузки, а синхронизация средства ведения помех велась с атакуемым устройством по сигналам CLK и DAT0 контроллера eMMC flash памяти устройства.

Показательным примером эффективности атак внедрения помех также является исследование [3]. В работе описан проведенный анализ устойчивости бортовых систем автомобиля Tesla к атакам посредством внедрения помех. Рассмотрены такие бортовые подсистемы, как информационно-развлекательная подсистема, подсистема автоматического управления автомобилем (автопилот) и шлюз безопасности шины CAN. Авторы отмечают, что, несмотря на грамотную реализацию механизмов безопасности данных, подсистема, компрометация информационно-развлекательной подсистемы и подсистемы автоматического управления

автомобилем возможна с помощью внедрения помех на линии питания, а компрометация шлюза безопасности шины CAN возможна с помощью электромагнитного внедрения помех.

В источнике [4] описаны исследования по применению электромагнитного внедрения помех для реализации повышения привилегий в операционной системе устройства Google TV Streamer. В источнике представлена полная последовательность действий по анализу встраиваемого устройства от распаковки и первичного анализа встраиваемого ПО до обхода проверок безопасности в системном вызове `setresuid` ядра ОС Android с помощью электромагнитного внедрения помех. Атакуемая система использует процессор Mediatek MT8696 и механизмы безопасности, присущие классическим устройствам с ОС Android, такие как Android Verified Boot, SELinux и доверенную среду исполнения на основе технологии ARM TrustZone. Тем не менее устройство также уязвимо к атакам посредством электромагнитного внедрения помех. Стоит отметить, что реализация электромагнитного внедрения помех в данное устройство возможно без модификации состава его аппаратного и программного обеспечения.

Все описанные исследования объединяет использование физической несовершенности встраиваемых систем для нейтрализации реализованных в них механизмов безопасности. В связи с тем, что данная несовершенство свойственна всем встраиваемым системам, особенно актуальной является проблема защиты данных систем от атак посредством внедрения помех. Для разработки новых методов устранения данной проблемы необходимо систематизировать каналы внедрения помех во встраиваемые системы, а также провести анализ современных средств внедрения помех и выделить их основные особенности. Современные работы по систематизации в данной области [5–8] сосредоточены на систематизации конкретных атак, а не каналов их реализации. При этом достоинства и недостатки средств проведения атак, важные для реализации новых методов и средств защиты от атак

данного рода, в данных работах не рассматриваются. Целью исследования является восполнение теоретического пробела в данной области и формирование теоретической базы для создания новых методов и средств защиты от атак на встраиваемые системы посредством внедрения помех.

2. ОПРЕДЕЛЕНИЕ И ОСНОВЫ ВНЕДРЕНИЯ ПОМЕХ

Перед систематизацией исследований в области внедрения помех необходимо обозначить определение помех. По данным ГОСТ Р 50397-2011, помеха – явление или процесс, которые снижают или могут снизить качество функционирования технического средства (ТС). В свою очередь ГОСТ 30804.6.2-2013 вводит следующие критерии качества функционирования ТС:

- требуется функционирование ТС без отклонения от требований технических условий при внедрении помех;
- требуется функционирование ТС без отклонения от требований технических условий после завершения процесса внедрения помех, но допускается ухудшение качества функционирования ТС во время внедрения помех;
- допускается временное прекращение выполнения функции ТС во время внедрения помех.

В контексте информационной безопасности внедрение помех зачастую рассматривается как процесс влияния злоумышленника на атакуемую встраиваемую систему и изменения ее поведения для нейтрализации реализованных в ней механизмов безопасности. Источник [9] приводит следующие ошибки работы встраиваемых систем, к которым может привести внедрение помех:

- инверсия бита (Bit Flip) – изменение значения одиночного бита в ячейке памяти на противоположное (с 0 на 1 или наоборот);
- установка/сброс значения бита (Bit set/Bit reset) – изменение значения одиночного бита в ячейке памяти на «1» (Bit set) или «0» (Bit reset);

- случайная мутация байта (Random byte) – изменение значения произвольного байта в ячейке памяти на другое, выбранное случайным образом;

- пропуск инструкции (Instruction skip) – отсутствие факта исполнения определенных процессорных инструкций;

- ошибки исполнения (Execution faults) – комплексные ошибки, состоящие из комбинаций других ошибок;

- константные неисправности (Stuck-at-faults) – фиксирование состояния определенных выходных сигналов сложно-функциональных блоков аппаратного обеспечения на заданном логическом уровне.

Описанные ошибки возникают за счет следующих факторов:

- переход одиночных логических сигналов сложно-функциональных блоков в состояния, не эквивалентные логическим «0» и «1» (состояние метастабильности);

- нарушение принципа синхронности сложно-функциональных блоков;

- уменьшение скорости распространения сигнала внутри сложно-функционального блока.

3. СИСТЕМАТИЗАЦИЯ КАНАЛОВ ВНЕДРЕНИЯ ПОМЕХ

Атаки на встраиваемые системы посредством внедрения помех заключаются в манипуляции физическими параметрами атакуемой системы, такими как тактовый сигнал, напряжение на линии питания или окружающее электромагнитное поле. Для манипуляции данными параметрами обычно используются каналы встраиваемой системы, например, линия питания или линия передачи тактового сигнала. Поэтому целесообразно систематизировать именно данные каналы. В качестве критериев систематизации выбраны физическая природа канала, степень инвазивности и сложность реализации атак внедрения помех по данным каналам.

Внедрение помех на линии передачи тактового сигнала. В современных интегральных схемах, использующих последовательную логику, синхронизация

между сложно-функциональными блоками и внутри сложно-функциональных блоков происходит посредством применения специального сигнала, называемого тактовым сигналом [10]. Данный сигнал имеет форму меандра и является цифровым, т.е. множество его значений принадлежит диапазону [0; 1]. Тактовый сигнал имеет следующие основные характеристики [11]:

- период – длина такта, время между одинаковыми соседними импульсами;

- частота – величина, обратная периоду и описывающая количество тактов за единицу времени;

- скважность – величина, описывающая отношение ширины высокого уровня сигнала к периоду такта.

В области внедрения помех тактовый сигнал зачастую рассматривается как примитив синхронизации фаз выполнения конвейера, отвечающего за декодирование, исполнение и обработку результатов процессорных инструкций.

Как правило, для проведения атак на встраиваемые системы посредством внедрения помех по тактовому сигналу применяют два основных подхода к синтезу помех: изменение длительности одиночных тактов и добавление одиночных тактов в последовательность.

Данные модификации последовательности импульсов тактового сигнала вызывают в аппаратном обеспечении нарушения принципа синхронности сложно-функциональных блоков, зачастую приводят к пропуску процессорных инструкций и используются для предотвращения исполнения инструкций условных переходов, например, при проверке целостности загрузчиков в контексте механизма доверенной загрузки.

Основными достоинствами данного канала внедрения помех являются воздействие на всю используемую интегральной схемой логику и сложность компенсации на стороне атакуемой системы, а основными недостатками – риск неконтролируемых последствий и ограниченная применимость в системах с использованием механизмов многофазовой синхронизации и автоподстройки частоты.

Внедрение помех на линии питания. Современные встраиваемые системы обычно имеют ряд требований, накладывающих ограничения на выбор компонентов данных систем: требования по физическим размерам аппаратного обеспечения, максимальному энергопотреблению, времени автономной работы, себестоимости и т.д. Для удовлетворения данных требований на этапе проектирования встраиваемых систем в архитектуру закладываются некоторые компромиссы. Одним из таких компромиссов является недостаточная стабилизация линии питания ядер микропроцессора. Поскольку современные интегральные схемы используют последовательную логику, отсутствие стабилизации линии питания ядер микропроцессора приводит к тому, что изменения напряжения на данной линии могут привести к ошибкам работы микропроцессора [12].

Атаки внедрения помех на линии питания микропроцессора имеют следующие основные этапы:

- предварительный анализ атакуемой системы: поиск информации об атакуемой системе в открытых источниках; анализ атакуемого ПО и поиск его фрагментов, подлежащих атаке; поиск примитивов синхронизации средства проведения атаки с атакуемой системой;
- удаление стабилизирующих конденсаторов с линии питания;
- подключение средства проведения атаки к линии питания, линии перезагрузки, а также линиям, содержащим примитивы синхронизации, атакуемой системы;
- проведение атаки: перебор параметров, приводящих к требуемому результату (например, пропуску инструкций);
- использование полученных при внедрении помех ошибок для выполнения поставленных задач.

Основными достоинствами линии питания встраиваемых систем как канала внедрения помех является относительная простота реализации атак и возможность синхронизации с атакуемой системой по трассам энергопотребления. Основным недостатком данного канала – частая необходимость физической модификации атакуемой системы.

Внедрение помех электромагнитным излучением. Атаки внедрения помех по рассмотренным каналам безусловно являются инвазивными. Однако существуют и неинвазивные атаки, связанные с внедрением помех посредством влияния на атакуемые системы электромагнитным излучением. Как известно из [13], в спектр электромагнитного излучения включают радиоволны, инфракрасное, световое, рентгеновское и гамма излучения. Некоторые из этих видов излучения способны вызывать в полупроводниковых материалах, используемых в составе интегральных схем, эффект ионизации – процесс перехода электронов из связанных состояний в свободные [14]. Рассмотрим основные применения электромагнитного излучения для внедрения помех во встраиваемые системы.

Индукционное внедрение помех. Внедрение помех во встраиваемые системы также возможно за счет изменения электромагнитного поля, окружающего данные системы. В основе данной техники внедрения помех лежит закон электромагнитной индукции Фарадея [15].

При прохождении импульсного или высокочастотного тока через катушку-зонд создается изменяющийся во времени магнитный поток, вызывающий появление индукционных токов в электрических контурах интегральной схемы встраиваемой системы [16]. Данный процесс приводит к переходному перенапряжению – явлению, заключающемуся в кратковременном всплеске напряжения, многократно превышающему номинальное. Это приводит к накоплению заряда на полупроводниковых элементах транзисторов и вызывает ошибки в исполнении ПО встраиваемой системы.

Основными достоинствами данного канала внедрения помех являются отсутствие инвазивности и простота реализации относительно других видов электромагнитного излучения. Главный недостаток данного канала – малая локальность. Ввиду того, что индукционное внедрение помех изменяет поле вокруг интегральной схемы, влияние на отдельные элементы данной схемы без влияния на соседние элементы очень затруднено.

Оптическое внедрение помех является относительно новой техникой, основная идея которой заключается в применении лазерных технологий для засвечивания транзисторов, входящих в состав интегральных схем. Внедрение помех таким образом происходит благодаря тому, что оптическое излучение может вызывать ионизацию полупроводниковых материалов, из которых выполнены транзисторы, если энергия фотонов превышает ширину запрещенной зоны данных материалов. Ионизация в данном случае получается благодаря процессу внутреннего фотоэффекта, при котором фотоны, выделяемые лазерной установкой, поглощаются, а их энергия тратится на перевод электронов полупроводникового материала из валентной зоны в зону проводимости.

В работе [17] описываются одни из первых исследований на тему оптического внедрения помех. Описана методика оптического внедрения помех и результаты ее применения для внедрения помех в микроконтроллер Microchip PIC16F84. Для внедрения помех использовалась фотовспышка Vivitar 550FD в сочетании с установкой для зондирования пластин Wentworth Laboratories MP-901. В результатах авторы обозначили возможность изменения произвольных битов SRAM памяти микроконтроллера со значения «0» на «1» и наоборот с помощью фокусировки света вспышки на элементах интегральной схемы, хранящих заряд, описывающий данные значения.

В качестве достоинств данного канала внедрения помех можно выделить его точность и предсказуемость работы. Основными недостатками являются падение точности с усложнением внутреннего строения интегральной схемы и высокая стоимость установок для проведения атак внедрения помех по данному каналу в современные интегральные схемы.

Внедрение помех с помощью рентгеновского излучения. Наиболее молодое направление исследований в области проведения атак на встраиваемые системы посредством внедрения помех связано с техниками внедрения помех с помощью направленного рентгеновского излуче-

ния. С точки зрения физического воздействия, рентгеновское излучение ионизирует полупроводниковые материалы и вызывает накопление заряда в диэлектрических слоях транзисторов и ячеек памяти. В источнике [17] описана возможность использования рентгеновского излучения как неинвазивного метода внедрения помех в связи с тем, что данный вид излучения может проходить сквозь металлические слои и материалы корпуса интегральных схем.

В исследовании [18] описан процесс проведения атаки посредством внедрения помех направленным рентгеновским излучением на интегральную схему, выполненную по 28-нанометровому процессу с применением технологии КМОП. Атака на данную микросхему проводилась с использованием европейского синхротрона ESRF – электровакуумной установки, способной создавать рентгеновское излучение посредством циклического ускорения частиц до скорости, близкой к скорости света. Данный синхротрон используется в работе для точечного излучения отдельных транзисторов, входящих в состав атакуемой интегральной схемы. Авторы отмечают, что в транзисторах, попадающих под влияние рентгеновского излучения, наблюдаются эффекты ионизации. Для транзисторов *p*-типа данные эффекты проявляются в виде закрытия, а для транзисторов *n*-типа – в виде открытия.

Учитывая все перечисленные особенности данного вида воздействия, наиболее важным достоинством канала внедрения помех с применением рентгеновского излучения является отсутствие требований к физической декапсуляции интегральной схемы перед проведением атаки. Один из наиболее существенных недостатков данного канала внедрения помех – требования к наличию дорогостоящего оборудования. Однако авторы источника [19] описывают реализацию атаки внедрения помех с использованием специализированной лабораторной установки, стоимость которой несравнимо меньше стоимости синхротрона ESRF. Тем не менее в результатах описано уменьшение точности воздействия на интегральную схему: от возможности

влияния на области схемы, содержащие отдельные биты, до возможности влияния на области схемы, содержащие группы из нескольких триггеров.

Программно-реализуемое внедрение помех. В контексте данной работы под программно-реализуемым внедрением помех понимается внедрение помех, в котором источником помех является не внешнее устройство, а внутреннее ПО, выполняющееся в атакуемой системе. Атаки программно-реализуемого внедрения помех, в отличие от рассмотренных, не требуют наличия физического доступа к атакуемому устройству, так как в качестве точки входа при проведении данных атак выступает ПО, исполняемое атакуемым устройством.

Основной особенностью данных атак является тот факт, что исполняемое атакуемой системой ПО (например, ядро операционной системы) может не содержать классических уязвимостей (переполнение буфера, *use-after-free* и др.), но при этом может быть использовано для внедрения помех посредством комбинации легитимных запросов от менее привилегированного ПО. Это связано с тем, что механизмы оптимизации исполнения программного кода, реализованные в аппаратном обеспечении, такие как применение кэша памяти и инструкций, а также суперскалярность, зачастую не учитываются при разработке прикладного ПО, выполняемого процессорами с реализацией данных механизмов. Рассмотрим примеры атак программно-реализуемого внедрения помех.

CLKScrew. В источнике [20] представлена работа по программно-реализуемому внедрению помех в устройство Google Nexus 6. Описывается атака CLKScrew, заключающаяся в использовании подсистемы DVFS (Dynamic Voltage and Frequency Scaling) процессора Qualcomm Snapdragon 805 для внедрения помех в работу доверенного ПО, исполняющегося в изолированной среде Trusted Execution Environment.

Подсистема DVFS предоставляет программному коду, исполняемому на уровне ядра ОС, функционал по изменению напряжения линии питания и множителей частоты тактового сигнала ядер процессора для реализации режимов энергосбере-

жения устройства. Авторы отмечают, что использование регуляторов, предоставляемых подсистемой DVFS для увеличения тактовой частоты или уменьшения напряжения линии питания процессора может привести к нарушению ограничений на время распространения сигнала в интегральных схемах.

Используются регуляторы подсистемы DVFS для реализации двух сценариев атак: внедрение помех при расшифровании информации алгоритмом AES для уменьшения пространства значений для перебора ключевого материала и при проверке криптографической подписи RSA образа прикладного ПО доверенной среды исполнения (трастлета). Оба сценария атак являются успешными. В случае с внедрением помех при расшифровании авторам удалось уменьшить пространство значений для перебора ключевого материала с 2^{127} до 2^{12} элементов благодаря внедрению помех при расшифровке отдельных байтов шифртекста и сравнению открытых текстов, полученных с внедрением помех при расшифровании и без внедрения помех.

В случае с внедрением помех при проверке криптографической подписи RSA авторам удалось внедрить помеху в работу функции *FlipEndianness* для изменения значения параметра N , являющегося модулем RSA, на значение N_a . Далее авторы описывают процесс создания подписи S'_a , валидной при использовании модуля N_a и внедряют созданную подпись в трастлет, проверка подписи которого будет успешной при использовании модуля N_a а затем загружают полученный трастлет, проводя внедрение помех при проверке его криптографической подписи. Таким образом, исследование показывает, что внутренние механизмы контроля тактовой частоты и напряжения линии питания процессора могут быть использованы для внедрения помех и нейтрализации механизмов безопасности операционной системы. Авторы подмечают, что при проведении описанной атаки изоляция доверенной среды исполнения не имела никакой важности, так как используемые примитивы распространялись и на реализацию доверенной среды исполнения.

GDDRHammer. В источнике [21] представлено исследование возможности модификации произвольных битов динамической оперативной памяти без прямого доступа к данным битам. Выдвинута гипотеза о том, что большое количество повторений доступа к двум областям виртуального адресного пространства процесса может изменить значение бита в третьей области памяти ЭВМ, даже если данная область расположена за границами виртуального пространства этого процесса. Такое поведение описывается большой плотностью современных интегральных схем типа DRAM и тем фактом, что при большом количестве обращений к одним и тем же ячейкам заряд может перетекать на соседние ячейки.

На листинге представлен фрагмент программного кода, вызывающего изменение значений битов DRAM памяти. Как можно заметить, исполняемый программный код осуществляет чтение из оперативной памяти по адресам X и Y в бесконечном цикле и сбрасывает кэш памяти для данных адресов.

Представленная атака получила продолжение в источнике [22]. Авторы описывают атаку на видеоускорители NVIDIA, использующие GDDR6 видеопамять. В отличие от предыдущей атаки, авторы атаки GDDRHammer проверяют гипотезу о том, что видеопамять GDDR6 имеет плотность не меньше, чем в интегральных схемах DRAM, и тоже подвержена изменению значения одиночных битов за счет повторяющихся доступов к ячейкам памяти.

Plundervolt. В источнике [23] описана атака посредством внедрения помех на ПО для процессоров Intel, использующее технологию Intel Software Guard Extensions (SGX) [24]. Данная технология пре-

доставляет системному и прикладному ПО, разрабатываемому для процессоров Intel, возможность создания анклавов – специальных областей виртуальной памяти, предназначенных для хранения конфиденциальной информации и программного кода, реализующего логику обработки данной информации. В результате анализа интерфейсов взаимодействия с данными процессорами выявлены незадокументированные моделезависимые системные регистры (MSR) с идентификаторами 0x150 и 0x198. Изменение содержимого регистра 0x150 (например, с помощью модуля *msr* ядра ОС семейства Linux) позволяло изменять напряжение на линии питания компонентов центрального процессора. Чтение содержимого регистра с идентификатором 0x198 позволяло определить текущее напряжение линий питания компонентов процессора. Взаимодействие с данными регистрами предоставляло примитивы по внедрению помех. Полученные примитивы использовались для изменения результатов выполнения ассемблерных инструкций, реализующих операцию умножения, таких как *mul* и *imul*. По словам авторов, процесс атаки имеет следующие основные этапы:

- ожидание выполнения атакуемой инструкции во время операции с конфиденциальной информацией внутри SGX анклава (например, расшифрование сообщения с помощью симметричной крипто-системы);
- уменьшение напряжения на линии питания процессора;
- получение результата операции, проведенной во время внедрения помех;
- сравнение полученного результата с результатом операции, проведенной без внедрения помех.

Фрагмент программного кода, реализующего изменения значений соседних битов DRAM памяти

```
code1a:
    mov (X), %eax
    mov (Y), %ebx
    clflush (X)
    clflush (Y)
    mfence
    jmp code1a
```

Атака, представленная в работе, отличается от классических атак внедрения помех по линии питания тем, что внедрение помех происходит неинвазивно и реализуется из контекста привилегированного ПО, исполняемого на атакуемой системе.

Выводы. Результаты систематизации каналов проведения атак на встраиваемые системы посредством внедрения помех указывают на разнообразие данных каналов. Однако, несмотря на данное разнообразие, все каналы объединяет нарушение синхронизации между сложно-функциональными блоками в результате внедрения помех. Наиболее опасным с точки зрения проведения атак на встраиваемые системы каналом внедрения помех является линия питания атакуемой системы. Это связано со сравнительной простотой реализации атак по данному каналу и низкой стоимостью используемого оборудования.

Стоит отметить, что ранние работы по систематизации [5–8] относили оптическое и рентгеновское внедрение помех к отдельным видам. Однако результаты данной работы указывают, что ввиду того, что используемые при внедрении помех виды излучения являются подвидами электромагнитного излучения, то и каналы внедрения помех тоже целесообразно относить к электромагнитному излучению.

4. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ ПРОВЕДЕНИЯ АТАК НА ВСТРАИВАЕМЫЕ СИСТЕМЫ ПОСРЕДСТВОМ ВНЕДРЕНИЯ ПОМЕХ

Первые исследования по проведению атак внедрения помех использовали отладочные платы общего назначения для реализации атак. Однако с развитием данной области появились специализированные средства, предназначенные для реализации атак данного класса. Рассмотрим основные представители данного класса устройств.

ChipWhisperer HuskyPlus. В источнике [25] впервые описано средство ChipWhisperer, его представляют как модульную платформу для проведения атак

посредством внедрения помех по линии питания и линии тактового сигнала, а также для проведения анализа побочных каналов встраиваемых систем. Основной упор делался на обеспечение доказательств реализуемости атак внедрения помех. Архитектура предложенного средства относительно простая, а само средство заточено на лабораторное применение и образовательные проекты.

Представленное средство доработано до версии ChipWhisperer HuskyPlus [26]. Оно предоставляет не только возможность проведения атак посредством внедрения помех на линии питания и линии передачи тактового сигнала, но и возможность проведения анализа побочных каналов, а также реализует поддержку трассировки исполнения микропроцессоров архитектуры ARM (механизм TraceWhisperer). Одной из наиболее примечательных особенностей средства ChipWhisperer HuskyPlus (рис. 1) является возможность внедрения помех на линии передачи тактового сигнала с наносекундным разрешением.

Основными достоинствами ChipWhisperer HuskyPlus являются: развитая экосистема, большое количество документации на тему использования средства, поддержка продвинутых методов анализа сторонних каналов и возможность внедрения помех во время трассировки микропроцессоров архитектуры ARM. В качестве основных недостатков данного средства можно выделить высокую стоимость и сложность аппаратного обеспечения, а также закрытость некоторых компонентов средства.

Chip'olino. В источнике [27] представлено средство под названием Chip'olino, являющееся разработкой эксперта R&D департамента Positive Labs компании Positive Technologies [28]. Платформа позиционируется автором как средство проведения атак на встраиваемые системы за счет внедрения помех по линии питания. В отличие от некоторых аналогов, Chip'olino подразумевает использование специальных плат расширения, предназначенных для подключения атакуемых микроконтроллеров к средству и называемых аддонами (рис. 2).

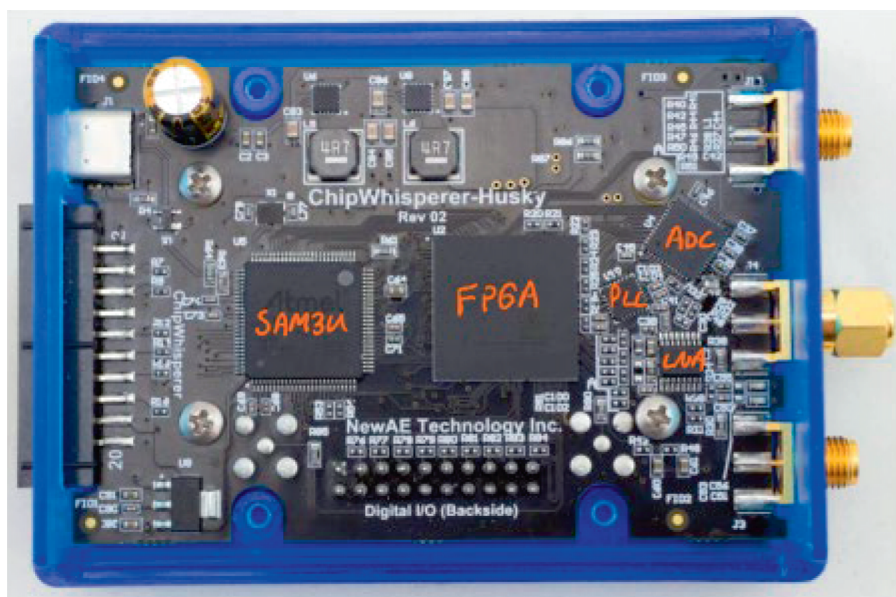


Рис. 1 Внешний вид средства ChipWhisperer HuskyPlus [26]

Fig. 1 | Appearance of the ChipWhisperer HuskyPlus [26]

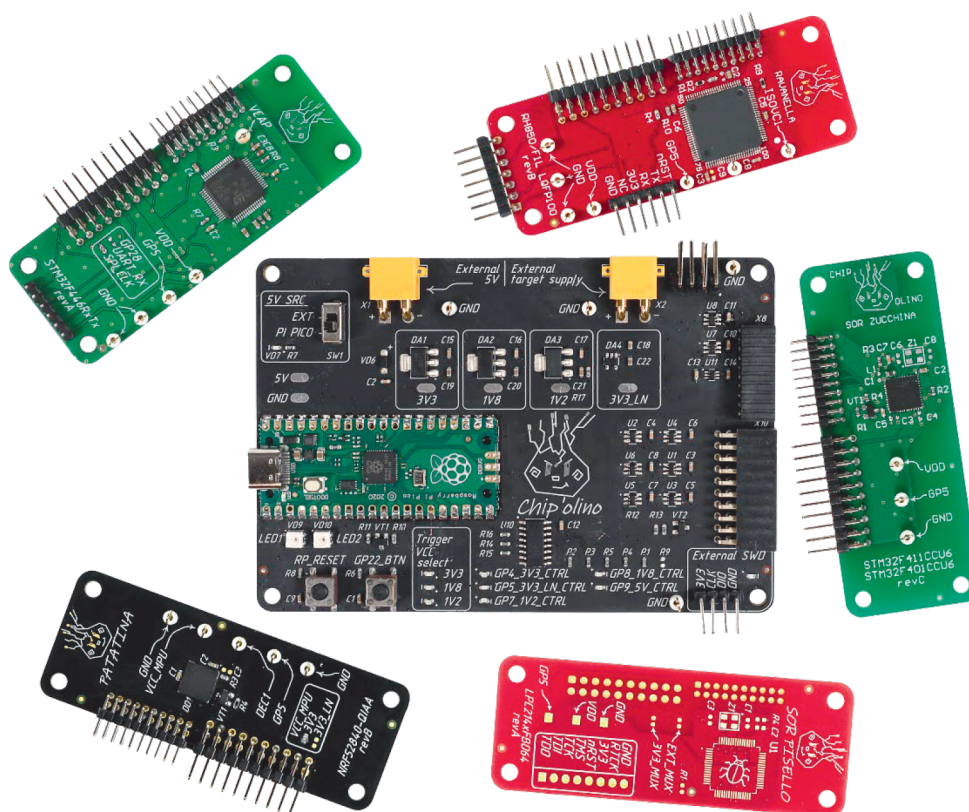


Рис. 2 Внешний вид печатной платы средства Chip'olino и плат расширения для нее [27]

Fig. 2 | The appearance of the Chip'olino printed circuit board and its expansion boards [27]

Основными достоинствами Chip'olino являются открытость и относительная простота аппаратного обеспечения, возможность взаимодействия с атакуемым устройством по протоколу Serial Wire Debug (SWD) и высокая переносимость условий атаки между экземплярами средства. В качестве основного недостатка можно выделить возможность повреждения атакуемого микроконтроллера при пересадке на плату расширения.

Ledger Donjon Scaffold. В источнике [29] представлено средство Scaffold (рис. 3), которое предназначено для проведения атак с помощью внедрения помех на линии питания, линии тактового сигнала, а также с применением электромагнитной индукции (в сочетании со средством SiliconToaster [30]). Данное средство построено на базе ПЛИС Altera Cyclone IV и имеет поддержку взаимодействия с атакуемыми системами по протоколам UART, SPI, I2C и ISO7816.

Как и средство Chip'olino, Scaffold подразумевает использование плат расширения, однако платы средства Scaffold имеют большую вариативность функционала, ввиду их подключения к интерфейсам ПЛИС.

Управление средством Scaffold осуществляется по интерфейсу USB с помощью ПО, написанного на языке программирования Python.

Основными достоинствами средства Scaffold являются возможность взаимодействия с атакуемой системой по большому количеству различных протоколов, поддержка внедрения помех на линии передачи тактового сигнала, а также использование аналогового усилителя для анализа побочных каналов. Основным недостатком данного средства – малое количество документации и открытых примеров использования средства.

PicoGlitcher. В источнике [31] представлено средство PicoGlitcher, в основе которого используется модуль Raspberry Pi Pico 2 на базе микроконтроллера Raspberry Pi RP2350. К данному модулю подключена пара металл-оксид-полупроводниковых транзисторов (MOSFET) для кратковременного понижения напряжения на линии питания атакуемого устройства. На рис. 4 представлен внешний вид печатной платы средства внедрения помех PicoGlitcher версии 3.

Основными достоинствами средства PicoGlitcher версии 3 являются возмож-

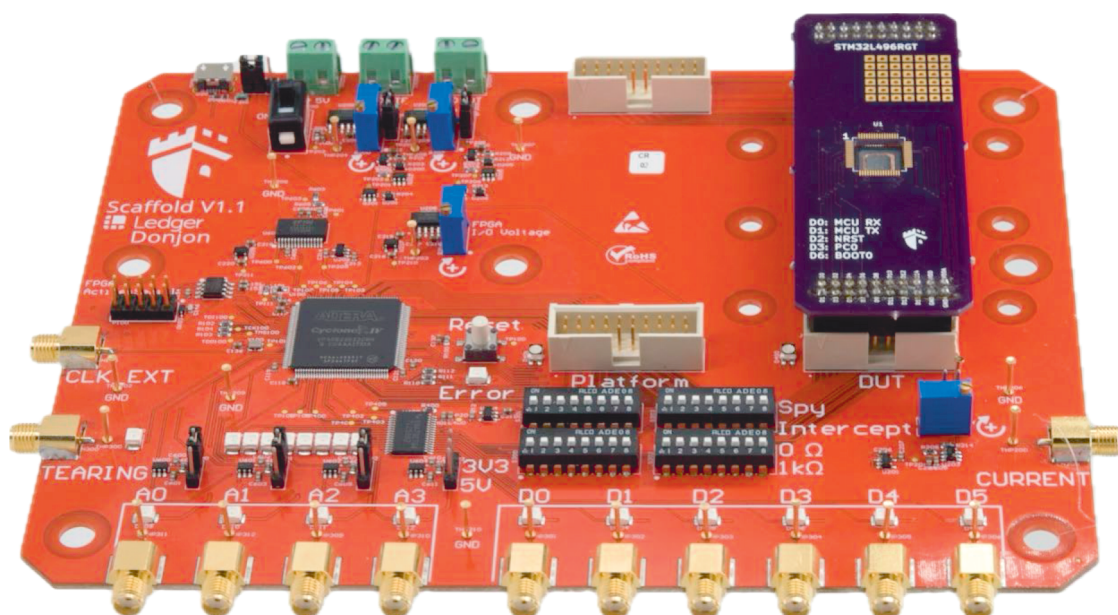


Рис. 3 | Внешний вид печатной платы средства Scaffold [29]

Fig. 3 | Appearance of the Scaffold [29] printed circuit board

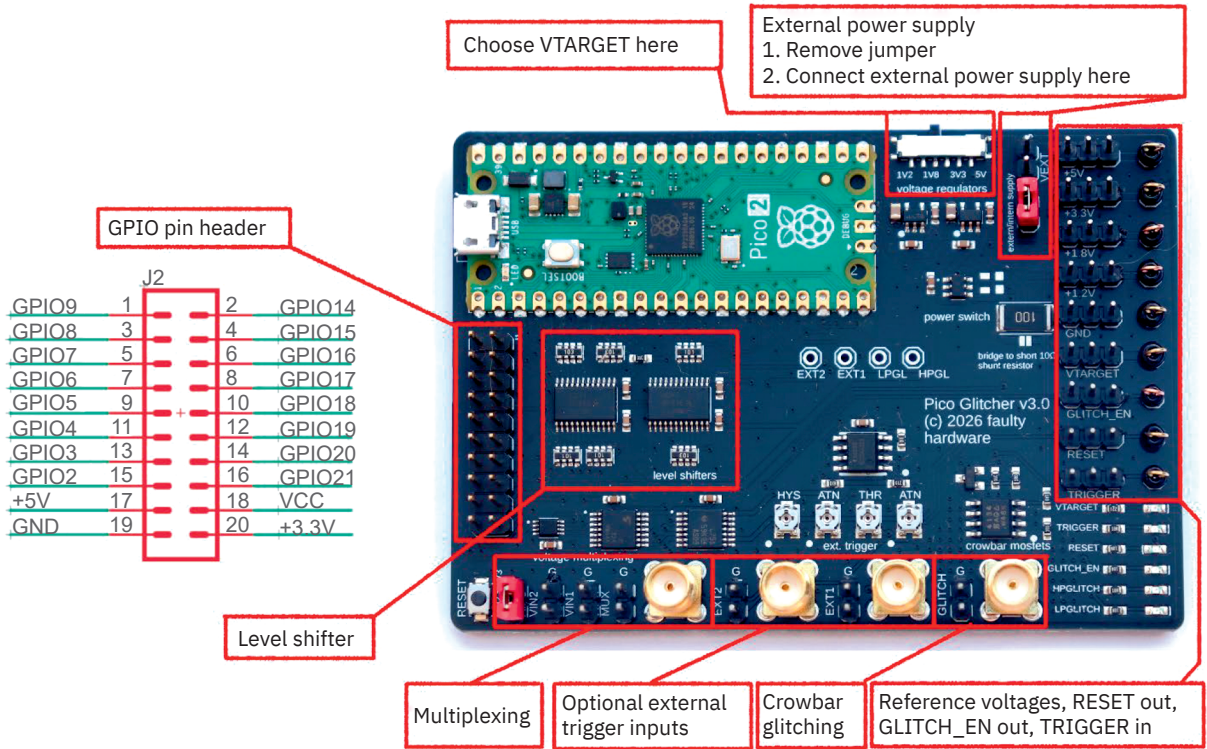


Рис. 4 | Внешний вид печатной платы средства PicoGlitcher версии 3 [31]

Fig. 4 | Appearance of the printed circuit board of the PicoGlitcher device, version 3 [31]

ности фильтрации входных сигналов от шума с помощью триггеров Шмитта и формирования напряжения на линии питания с помощью цифрового синтеза. Основными недостатками средства являются отсутствие функционала по внедрению помех на линии тактового сигнала и закрытость аппаратного обеспечения.

Сравнение рассмотренных средств. В результате сравнения рассмотренных средств (см. таблицу) выявлено, что средства с большим количеством различного дополнительного функционала имеют в качестве управляющего модуля программируемые логические интегральные схемы. Также выявлено, что полностью открытыми и доступными для воспроизведения являются Scaffold [31] и отечественная разработка Chip'olino [27].

Выводы. Учитывая результаты проведенного сравнительного анализа средств проведения атак на встраиваемые системы посредством внедрения помех, можно сделать вывод о том, что рассмотренные

средства имеют высокую эффективность работы по заявлениям разработчиков и используют электронные компоненты со сравнительно низкой стоимостью и легкодоступные для получения в больших количествах. Таким образом, современные средства проведения атак на встраиваемые системы посредством внедрения помех не только легко воспроизводимы квалифицируемыми атакующими, но и предоставляют возможность своей модификации и адаптации под конкретные атакуемые системы.

5. ЗАКЛЮЧЕНИЕ

Проведена систематизация каналов, а также выполнен сравнительный анализ средств проведения атак посредством внедрения помех, исследованы их достоинства и недостатки, выделены принципы их работы.

Сравнение функциональных возможностей средств внедрения помех
Comparison of the functional capabilities of interference injection tools

Критерий	Средство			
	ChipWhisperer HuskyPlus [26]	Chip'olino [27]	Scaffold v1.1 [29]	PicoGlitcher v3 [31]
Управляющий модуль	ПЛИС Xilinx Artix 7	МК RP2040	ПЛИС ALTERA Cyclone IV	МК RP2350
Поддерживаемые каналы внедрения помех	Линия питания, линия передачи тактового сигнала	Линия питания	Линия питания, линия передачи тактового сигнала, электромагнитное излучение (в сочетании с SiliconToaster [30])	Линия питания
Открытость программного и аппаратного обеспечения	Частично открыты (аппаратное обеспечение закрыто)	Полностью открыты	Полностью открыты	Частично открыты (аппаратное обеспечение закрыто)
Дополнительные функциональные возможности	Наличие встроенного 12-битного АЦП; возможность проведения анализа побочных каналов; поддержка TraceWhisperer	Поддержка использования внешнего источника питания; возможность взаимодействия с атакуемой системой по протоколу SWD; возможность применения плат-адаптеров для подключения атакуемой системы	Возможность взаимодействия с атакуемой системой по протоколам UART, I2C, ISO7816, SPI; наличие аналогового усилителя для с частотой выборки 200 МГц для анализа трасс энергопотребления; возможность применения плат-адаптеров для подключения атакуемой системы	Поддержка использования внешнего источника питания; возможность формирования напряжения на линии питания с помощью цифрового синтеза (в сочетании с Pulse Shaping Board [32])

Поскольку противодействие данным атакам является актуальной научно-технической задачей ввиду их сравнительно малой инвазивности и возможности полной компрометации механизмов безопасности атакуемых систем, в качестве направления дальнейших исследований

можно выделить направления, заключающиеся в оценке устойчивости встраиваемого ПО к атакам данного класса и повышению устойчивости за счет автоматического изменения фрагментов программного кода, уязвимого к атакам данного класса.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Glitched on Earth by Humans: A Black-Box Security Evaluation of the SpaceX Starlink User Terminal. URL: <https://blackhat.com/us-22/briefings/schedule/index.html?glitched-on-earth-by-humans-a-black-box-security-evaluation-of-the-spacex-starlink-user-terminal-26982> (дата обращения: 10.07.2026).
2. There's a hole in your SOC: Glitching the Mediatek BootROM Terminal. URL: <https://www.nccgroup.com/research/there-s-a-hole-in-your-soc-glitching-the-mediatek-bootrom/> (дата обращения: 10.07.2026).
3. Kühnapfel N., Werling C., Jacob H. N., Seifert J.-P. Three Glitches to Rule One Car: Fault Injection Attacks on a Connected EV // 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25). Association for Computing Machinery, New York, NY, USA. 2025. P. 1235–1249.
4. Glitching Google's TV Streamer: from ADB to root. URL: <https://hardwear.io/wp-content/uploads/2026/03/Glitching-Google's-TV-Streamer-from-adb-to-root.pdf> (дата обращения: 10.07.2026).
5. Shuvo A. M., Zhang T., Farahmandi F., Tehranipoor M. A Comprehensive Survey on Non-Invasive Fault Injection Attacks // Cryptology ePrint Archive. 2023. URL: <https://eprint.iacr.org/2023/1769.pdf> (дата обращения: 10.07.2026).
6. Gangolli A., Mahmoud Q. H., Azim A. A systematic review of fault injection attacks on iot systems // Electronics. 2022. Vol. 11. № 13. P. 2023.
7. Boulifa R., Di Natale G., Maistri P. Countermeasures Against Fault Injection Attacks in Processors: A Review // Information. 2025. Vol. 16. № 4. P. 293.
8. Yuce B., Schaumont P., Witteman M. Fault Attacks on Secure Embedded Software: Threats, Design and Evaluation // arXiv:2003.10513. 2020. DOI: 10.48550/arXiv.2003.10513.
9. Liu C. S., Gould P., Wang F., Yagemann C. SoK: A Beginner-Friendly Introduction to Fault Injection Attacks // arXiv:2509.18341. 2025.
10. Цифровая схемотехника. URL: <https://iu4.ru/edu/211001/sem08/dv1/lec.pdf> (дата обращения: 12.07.2026).
11. Языки описания схем. URL: https://mk.cs.msu.ru/images/5/5f/Hdl_lecture_07.pdf (дата обращения: 12.07.2026).
12. Designing with microcontrollers in noisy environments. URL: https://www.st.com/resource/en/application_note/cd00003833-designing-with-microcontrollers-in-noisy-environments-stmicroelectronics.pdf (дата обращения: 12.07.2026).
13. Литвинов О. С., Горелик В. С. Электромагнитные волны и оптика. М. : Изд-во МГТУ имени Н. Э. Баумана, 2006. 447 с.
14. Михайлова М. П., Иванов Э. В., Данилов Л. В. и др. Излучательная рекомбинация и ударная ионизация в полупроводниковых наноструктурах (Обзор) // Физика и техника полупроводников. 2020. Т. 54. № 12. С. 1267–1288.

15. Методическая разработка по общему физическому практикуму. Лаб. работа № 63. Электромагнитная индукция. Закон Фарадея. URL: <https://condmatt.phys.msu.ru/upload/iblock/fc8/t3d4sbgap3tpm3oh74rx1pc8skfpvlsf/363.pdf> (дата обращения: 14.07.2026).
16. **Moro N., Dehbaoui A., Heydemann K. et al.** Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller // 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography. 2013. P. 77–88.
17. **Skorobogatov S. P., Anderson R. J.** Optical fault induction attacks // International workshop on cryptographic hardware and embedded systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. P. 2–12. DOI: 10.1007/3-540-36400-5_2.
18. **Bouat S., Anceau S., Maingault L. et al.** X ray nanoprobe for fault attacks and circuit edits on 28-nm integrated circuits // 2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Oct 2023, Juan-Les-Pins, France. 2023. P. 1–6. DOI: 10.1109/DFT59622.2023.10313553.
19. **Maingault L., Anceau S., Sulmont M. et al.** Laboratory X-rays Operando Single Bit Attacks on Flash Memory Cells // Smart Card Research and Advanced Applications. CARDIS 2021. Lecture Notes in Computer Science. Vol. 13173. Springer, Cham. 2022. DOI: 10.1007/978-3-030-97348-3_8.
20. **Tang A., Sethumadhavan S., Stolfo S.** CLKSCREW: Exposing the perils of Security-Oblivious energy management // 26th USENIX Security Symposium (USENIX Security 17). 2017. P. 1057–1074.
21. **Kim Y., Daly R., Kim J. et al.** Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors // ACM SIGARCH Computer Architecture News. 2014. Vol. 42. № 3. P. 361–372.
22. **Hu Y., Brown N., Chen Yu. et al.** GDDR-Hammer: Greatly Disturbing DRAM Rows – Cross-Component Rowhammer Attacks From Modern GPUs // 2026 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. 2026. P. 546–564. DOI: 10.1109/SP63933.2026.00228.
23. **Murdock K., Oswald D., Garcia F. D. et al.** Plundervolt: Software-based fault injection attacks against Intel SGX // 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. 2020. P. 1466–1482. DOI: 10.1109/SP40000.2020.00057.
24. Intel. Intel Software Guard Extensions overview. URL: <https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html> (дата обращения: 16.07.2026).
25. **O’Flynn C., Chen Z.** Chipwhisperer: An open-source platform for hardware embedded security research // International Workshop on Constructive Side-Channel Analysis and Secure Design. Cham: Springer International Publishing, 2014. P. 243–260.
26. ChipWhisperer Husky & HuskyPlus manual. URL: https://media.newae.com/manuals/cwhusky_cwhuskyplus_manual_may2025.pdf (дата обращения: 12.07.2026).
27. y0v1737/chipolino. URL: <https://github.com/y0v1737/chipolino> (дата обращения: 18.07.2026).
28. Positive Research: Chip’olino: инструмент исследователя и домохозяйки. URL: <https://ptresearch.media/articles/chip-olino-instrument-issledovatelya-i-domohozyajki> (дата обращения: 18.07.2026).
29. Ledger-Donjon/scaffold. URL: <https://github.com/Ledger-Donjon/scaffold/> (дата обращения: 18.07.2026).
30. Ledger-Donjon/silicon-toaster. URL: <https://github.com/Ledger-Donjon/silicon-toaster> (дата обращения: 18.07.2026).
31. Overview. URL: <https://fault-injection-library.readthedocs.io/en/latest/overview/> (дата обращения: 18.07.2026).
32. Pulse-Shaping. URL: <https://fault-injection-library.readthedocs.io/en/latest/pulse-shaping/> (дата обращения: 20.07.2026).

REFERENCES

1. Glitched on Earth by Humans: A Black-Box Security Evaluation of the SpaceX Starlink User Terminal. URL: <https://blackhat.com/us-22/briefings/schedule/index.html?glitched-on-earth-by-humans-a-black-box-security-evaluation-of-the-spacex->

- starlink-user-terminal-26982 (accessed: 10.07.2026).
2. There's a hole in your SOC: Glitching the Mediatek BootROM Terminal. URL: <https://www.nccgroup.com/research/there-s-a-hole-in-your-soc-glitching-the-mediatek-bootrom/> (accessed: 10.07.2026).
 3. **Kühnapfel N., Werling C., Jacob H. N., Seifert J.-P.** Three Glitches to Rule One Car: Fault Injection Attacks on a Connected EV. 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25). Association for Computing Machinery, New York, NY, USA. 2025, pp. 1235–1249.
 4. Glitching Google's TV Streamer: from ADB to root. URL: <https://hardwear.io/wp-content/uploads/2026/03/Glitching-Google-TV-Streamer-from-adb-to-root.pdf> (accessed: 10.07.2026).
 5. **Shuvo A. M., Zhang T., Farahmandi F., Tehranipour M.** A Comprehensive Survey on Non-Invasive Fault Injection Attacks. Cryptology ePrint Archive. 2023. URL: <https://eprint.iacr.org/2023/1769.pdf> (accessed: 10.07.2026).
 6. **Gangolli A., Mahmoud Q. H., Azim A.** A systematic review of fault injection attacks on IoT systems. *Electronics*. 2022. Vol. 11. No. 13, pp. 2023.
 7. **Boulifa R., Di Natale G., Maistri P.** Countermeasures Against Fault Injection Attacks in Processors: A Review. *Information*. 2025. Vol. 16. No. 4, pp. 293.
 8. **Yuce B., Schaumont P., Witteman M.** Fault Attacks on Secure Embedded Software: Threats, Design and Evaluation. arXiv:2003.10513. 2020. DOI: 10.48550/arXiv.2003.10513.
 9. **Liu C. S., Gould P., Wang F., Yagemann C.** SoK: A Beginner-Friendly Introduction to Fault Injection Attacks. arXiv:2509.18341. 2025.
 10. Digital circuit design. URL: <https://iu4.ru/edu/211001/sem08/dv1/lec.pdf> (accessed: 12.07.2026). (In Russian)
 11. Schema description languages. URL: https://mk.cs.msu.ru/images/5/5f/Hdl_lecture_07.pdf (accessed: 12.07.2026). (In Russian)
 12. Designing with microcontrollers in noisy environments. URL: https://www.st.com/resource/en/application_note/cd00003833-designing-with-microcontrollers-in-noisy-environments-stmicroelectronics.pdf (accessed: 12.07.2026).
 13. **Litvinov O. S., Gorelik V. S.** Electromagnetic Waves and Optics. Moscow : Publishing House of the Bauman Moscow State Technical University, 2006, 447 p. (In Russian)
 14. **Mikhailova M. P., Ivanov E. V., Danilov L. V. et al.** Radiative Recombination and Impact Ionization in Semiconductor Nanostructures (a review). *Fizika i tehnika poluprovodnikov*. 2020. Vol. 54. No. 12, pp. 1267–1288. (In Russian),
 15. Methodological development for the general physics workshop. Lab. work No. 63. Electromagnetic induction. Faraday's law. URL: <https://condmatt.phys.msu.ru/upload/iblock/fc8/t3d4sbgap3tpm3oh74rx1pc8skfpvlsf/363.pdf> (accessed: 14.07.2026). (In Russian)
 16. **Moro N., Dehbaoui A., Heydemann K. et al.** Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller. 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography. 2013, pp. 77–88.
 17. **Skorobogatov S. P., Anderson R. J.** Optical fault induction attacks. International workshop on cryptographic hardware and embedded systems. Berlin, Heidelberg : Springer Berlin Heidelberg, 2002, pp. 2–12. DOI: 10.1007/3-540-36400-5_2.
 18. **Bouat S., Anceau S., Maingault L. et al.** X ray nanoprobe for fault attacks and circuit edits on 28-nm integrated circuits. 2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Oct 2023, Juan-Les-Pins, France. 2023, pp. 1–6. DOI: 10.1109/DFT59622.2023.10313553.
 19. **Maingault L., Anceau S., Sulmont M. et al.** Laboratory X-rays Operando Single Bit Attacks on Flash Memory Cells. Smart Card Research and Advanced Applications. CARDIS 2021. Lecture Notes in Computer Science. Vol. 13173. Springer, Cham. 2022. DOI: 10.1007/978-3-030-97348-3_8.
 20. **Tang A., Sethumadhavan S., Stolfo S.** CLKSCREW: Exposing the perils of Security-Oblivious energy management. 26th USENIX Security Symposium (USENIX Security 17). 2017, pp. 1057–1074.
 21. **Kim Y., Daly R., Kim J. et al.** Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. *ACM SIGARCH Computer Architecture News*. 2014. Vol. 42. No. 3, pp. 361–372.
 22. **Hu Y., Brown N., Chen Yu. et al.** GDDR-Hammer: Greatly Disturbing DRAM Rows – Cross-Component Rowhammer Attacks From Modern GPUs. 2026 IEEE Symposium

- on Security and Privacy (SP), San Francisco, CA, USA. 2026, pp. 546–564. DOI: 10.1109/SP63933.2026.00228.
23. **Murdock K., Oswald D., Garcia F. D. et al.** Plundervolt: Software-based fault injection attacks against Intel SGX. 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. 2020, pp. 1466–1482. DOI: 10.1109/SP40000.2020.00057.
 24. Intel. Intel Software Guard Extensions overview. URL: <https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html> (accessed: 16.07.2026).
 25. **O’Flynn C., Chen Z.** Chipwhisperer: An open-source platform for hardware embedded security research. International Workshop on Constructive Side-Channel Analysis and Secure Design. Cham: Springer International Publishing, 2014, pp. 243–260.
 26. ChipWhisperer Husky & HuskyPlus manual. URL: https://media.newae.com/manuals/cwhusky_cwhuskyplus_manual_may2025.pdf (accessed: 12.07.2026).
 27. y0v1737/chipolino. URL: <https://github.com/y0v1737/chipolino> (accessed: 18.07.2026).
 28. Positive Research: Chip’olino: a tool for the researcher and the homemaker. URL: <https://ptresearch.media/articles/chip-olino-instrument-issledovatelya-i-domohozyajki> (accessed: 18.07.2026). (In Russian)
 29. Ledger-Donjon/scaffold. URL: <https://github.com/Ledger-Donjon/scaffold/> (accessed: 18.07.2026).
 30. Ledger-Donjon/silicon-toaster. URL: <https://github.com/Ledger-Donjon/silicon-toaster> (accessed: 18.07.2026).
 31. Overview. URL: <https://fault-injection-library.readthedocs.io/en/latest/overview/> (accessed: 18.07.2026).
 32. Pulse-Shaping. URL: https://fault-injection-library.readthedocs.io/en/latest/pulse_shaping/ (accessed: 20.07.2026).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ПАГУБА Григорий Юрьевич – аспирант, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: paguba_gyu@spbstu.ru
ORCID: 0009-0001-3593-1417

КИНЕСШЕВ Даниил Сергеевич – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: kineshev.ds@edu.spbstu.ru
ORCID: 0009-0003-5808-3267

КОНДАЧКОВ Егор Дмитриевич – аспирант, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: kondachkov_ed@spbstu.ru
ORCID: 0009-0004-1441-8771

PAGUBA Grigoriy Yu. – Postgraduate Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

KINESHEV Daniil S. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

KONDACHKOV Egor D. – Postgraduate Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29