АППАРАТНАЯ И ПРОГРАММНАЯ КРИПТОГРАФИЯ

ЗИЯ САРДАР (ZIA SARDAR), старший технический специалист
ААРОН АРЕЛЬЯНО (AARON ARELLANO), старший технический специалист
СТЮАРТ МЕРКЕЛЬ (STEWART MERKEL), технический специалист компании, Maxim Integrated

В этой статье, продолжающей цикл публикаций нашего журнала (см. ЭК8–9, 11 за 2020 г.), рассматриваются различия в реализации криптографических решений с помощью аппаратных средств и с использованием программного обеспечения, а также основные этапы безопасной загрузки подключенного устройства.

Современные криптографические алгоритмы могут быть реализованы с помощью специального криптографического оборудования или с помощью программного обеспечения, работающего на оборудовании общего назначения. По разным причинам для большинства задач лучшим решением является специальное криптографическое оборудование. В таблице перечислены причины, по которым аппаратные криптографические решения являются более предпочтительными.

БЕЗОПАСНАЯ ЗАГРУЗКА И СКАЧИВАНИЕ – ЧТО ЭТО И ПОЧЕМУ ЭТО ВАЖНО?

К повседневным устройствам интернета вещей относятся, например:

- домашние устройства: Wi-Fi-камеры, IoT-термостаты и детекторы дыма;
- медицинские устройства;
- портативные электронные устройства, фитнес-трекеры и умные часы;
- роботизированные манипуляторы на заводах.

Почти все эти устройства (см. рис. 1) используют загрузочную прошивку или загружаемые данные, которые получают доступ к интернету, из-за чего устройства подвергаются угрозе быть взломанными. Фактически загрузочная прошивка сохраняется в энергонезависимой памяти внутри устройства. Время от времени это программное обеспечение обновляется для исправления и улучшения некоторых функций, к которым относятся и новый алгоритм обнаружения нарушителей в случае использования Wi-Fi-камеры, и изменение угла наклона руки промышленного робота для более точного выполнения сварного шва.

Мы рассмотрим все шаги, необходимые для безопасной загрузки и установки новой прошивки в подключенное устройство.

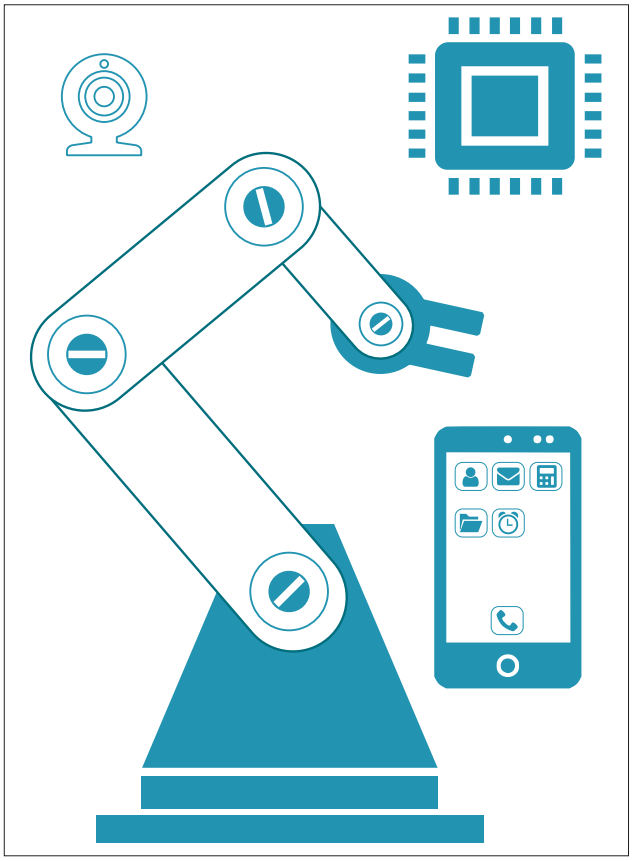

Рис. 1. **Устройства интернета вещей, например роботизированный манипулятор на заводе, имеют встроенное оборудование, использование которого может нарушить безопасность**

Таблица. **Сравнение аппаратной и программной криптографии**

Аппаратная криптография	Программная криптография
1. Используется специальное оборудование, значительно ускоряющее выполнение процесса	1. Используется оборудование общего назначения, из-за чего процессы выполняются медленнее
2. Не зависит от операционной системы. Работа оборудования поддерживается специализированным программным обеспечением	2. Зависит от уровней безопасности, возможностей операционной системы и поддерживаемого программного обеспечения
3. Может использовать заводскую инициализацию, безопасно хранить ключи и другие данные в выделенных защищенных ячейках памяти	3. Нет выделенных защищенных ячеек памяти. Из-за этого ключи и данные могут подвергаться краже или подделке
4. Устройства компании Maxim имеют встроенную защиту от реверс-инжиниринга, – физически неклонируемую функцию (ChipDNA)	4. Программные реализации проще подвергнуть реверс-инжинирингу
5. В аппаратной системе особое внимание уделяется сокрытию и защите важной информации (например, закрытых ключей), что позволяет сделать ее как можно более труднодоступной	5. При использовании системы общего назначения, в которой реализована программная криптография, существует много способов отслеживания и получения доступа к важной информации. Примером может служить перехват закрытого ключа при передаче внутри компьютерной системы

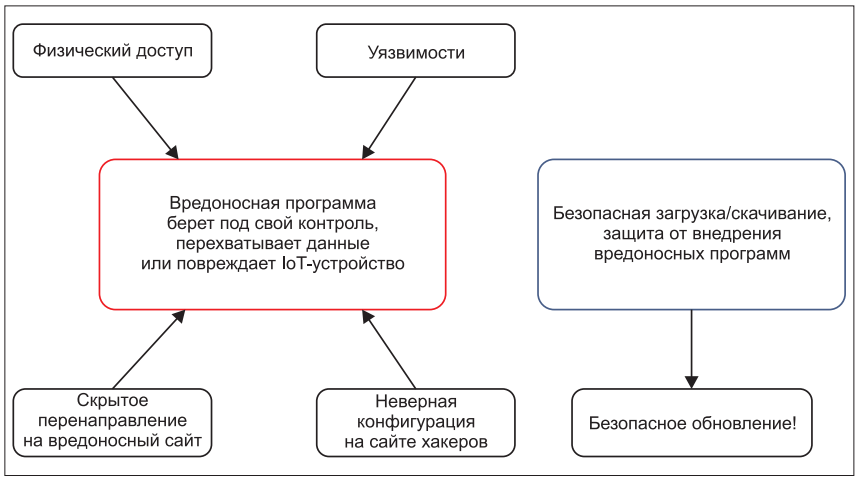

Рис. 2. **В отличие от защищенных IoT-устройств, незащищенные доступны для проникновения злоумышленников**

ЗАЧЕМ ЗАЩИЩАТЬ ПРОШИВКУ ИЛИ ДАННЫЕ IoT-УСТРОЙСТВ?

Устройства интернета вещей должны быть надежными, а это значит, что их прошивка и важные данные должны проверяться на подлинность. В идеальном случае загрузочная прошивка и файлы конфигурации защищены уже на заводе, но клиентам требуется, чтобы обновления прошивки и изменение настроек были доступны через интернет. В этом и заключается проблема – злоумышленники могут использовать сетевые интерфейсы в качестве канала передачи вредоносных программ.

Получив контроль над IoT-устройством, можно использовать его в своих целях. Поэтому, прежде чем использовать любой код, исходящий якобы из авторизованного источника, необходимо этот код аутентифицировать.

Злоумышленник может передать вредоносное ПО на IoT-устройство разными способами (см. рис. 2).

– Если злоумышленник получит физический доступ к устройству, он может ввести вредоносное ПО через физическое соединение (например, USB, Ethernet и т. д.).
– В операционных системах часто имеются уязвимости, по мере обнаружения которых выпускаются патчи. Если такие патчи вовремя не были установлены, злоумышленник получает доступ к незащищенной системе и внедряет вредоносное ПО.
– Часто IoT-устройства связываются с серверами обновлений, чтобы определить, доступны ли обновленные данные прошивки или конфигурации. Злоумышленник может перехватить DNS-запрос и перенаправить IoT-устройство на вредоносный источник с вредоносным ПО или искаженными файлами конфигурации.
– Подлинный сайт может быть неправильно сконфигурирован, что позволяет злоумышленнику взять его под свой контроль и заменить прошивку вредоносным кодом.

С помощью безопасной загрузки и скачивания мы можем предотвратить несанкционированный доступ и защититься от внедрения вредоносных программ. Таким образом, IoT-устройство сможет доверять обновлениям, получаемым из центра управления.

Обратите внимание: если центру управления потребуется доверять IoT-устройству, необходимо выполнить дополнительный шаг, включающий аутентификацию IoT-устройства. Как защитить эти устройства с помощью безопасной начальной загрузки и скачивания?

АУТЕНТИФИКАЦИЯ И ЦЕЛОСТНОСТЬ ПРОШИВКИ

Для аутентификации и целостности необходимо:

- убедиться, что устройство работает только с авторизованными прошивками или файлами конфигурации;
- проверить, что данные являются надежными и впоследствии не изменяются;
- использовать шифрование для подтверждения подлинности и целостности данных;
- использовать криптографические цифровые подписи (аналог рукописной подписи и печати, которые ставят в конце бумажных документов).

Для обеспечения подлинности и целостности данные прошивки и файлы конфигурации загружаются на заводе, а все последующие обновления имеют цифровую подпись. Таким образом, цифровая подпись обеспечивает безопасность в течение всего срока службы устройства. Для безопасности первостепенное значение имеют следующие аспекты цифровой подписи:

- используемая цифровая подпись создается с помощью криптографического алгоритма;
- для обеспечения самого высокого уровня безопасности используются общедоступные и хорошо зарекомендовавшие себя алгоритмы.

Далее мы рассмотрим асимметричные криптографические алгоритмы, в частности FIBS 186 ECDSA.

АСИММЕТРИЧНАЯ КРИПТОГРАФИЯ ДЛЯ БЕЗОПАСНОЙ ЗАГРУЗКИ/СКАЧИВАНИЯ

В асимметричной криптографии для реализации алгоритма используется пара из открытого и закрытого ключа (см. рис. 3).

Генерация любой пары ключей начинается с выбора случайного числа, которое будет использоваться в качестве закрытого ключа. Это случайное число вводится в генератор ключей для вычисления открытого ключа. Он становится

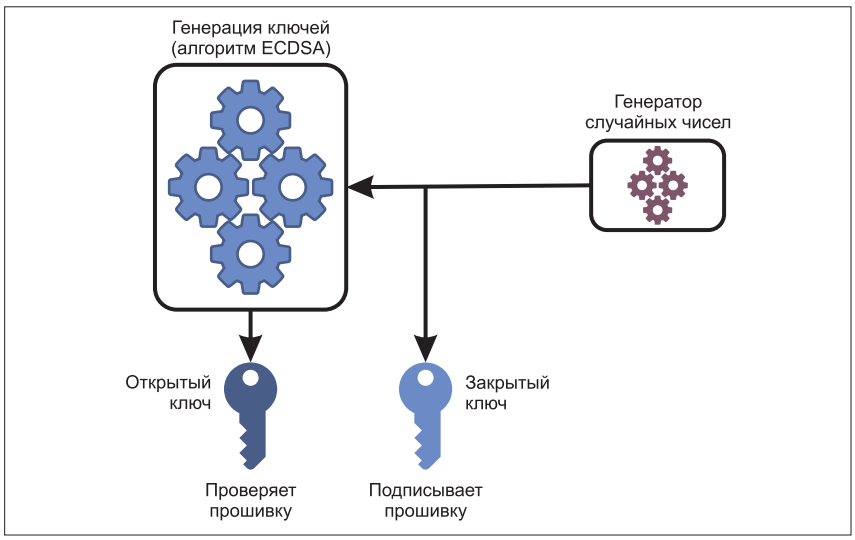

Рис. 3. **В асимметричной криптографии осуществляется генерация ключей по алгоритму ECDSA**

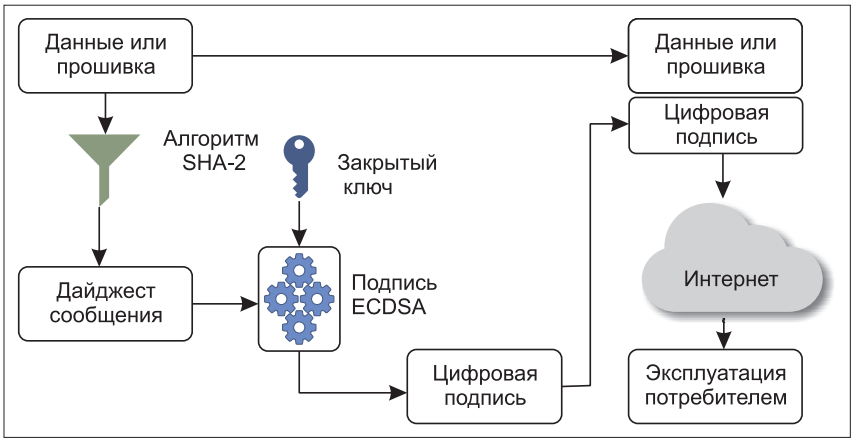

Рис. 4. **Цифровая подпись набора данных или прошивки при использовании асимметричной криптографии**

общедоступным (может свободно распространяться без какого-либо риска для безопасности). В отличие от него, закрытый ключ должен всегда храниться в тайне.

Основные принципы безопасной загрузки в асимметричной криптографии: разработчик прошивки использует для подписи закрытый ключ; встраиваемое устройство использует для проверки открытый ключ.

Зачем использовать криптографию с асимметричным ключом? Преимущество асимметричной криптографии заключается в том, что на устройстве не хранится закрытый ключ. При использовании асимметричной криптографии злоумышленник не может получить закрытый ключ. Наконец, выбранный алгоритм (например, ECDSA) делает математически невозможным получение закрытого ключа из открытого.

Для начала давайте рассмотрим пример того, что должно происходить на стороне разработчика и изготовителя, использующего асимметричную криптографию ключей.

На стороне разработчика

Начнем с прошивки. Она должна быть подвергнута многоблочному хэшированию SHA-256.

Закрытый ключ и хэш вводятся в алгоритм генерации подписи ECDSA. Результатом является уникальная подпись. Прошивка и подпись отправляются по запросу для использования в оборудовании. На рисунке 4 эти шаги показаны подробнее. Теперь давайте рассмотрим эксплуатацию оборудования у потребителя.

На стороне потребителя

Встроенное устройство получает прошивку и подпись. Прошивка проходит через многоблочное хэширование SHA-256. Наше встроенное устройство уже содержит открытый ключ, созданный во время генерации ключей на площадке разработчика. Подпись и другие элементы используются в качестве входных данных для проверки ECDSA. Результат проверки ECDSA определит, может ли встроенное устройство использовать эту прошивку. Если ответ положительный (т. е. подлинность и целостность подтверждены), встроенное устройство принимает прошивку. В случае отрицательного результата прошивка отклоняется.

БЕЗОПАСНОЕ СКАЧИВАНИЕ И БЕЗОПАСНАЯ ЗАГРУЗКА С ПОМОЩЬЮ DS28C36

Многие устройства не имеют защищенного микроконтроллера, предназначенного для проверки подлинности и целостности скачиваемых прошивок или данных. Одним из недорогих аппаратных решений является защищенное устройство для аутентификации DS28C36 DeepCover (см. рис. 5).

Шаги, необходимые для безопасной загрузки:

1. На этапе разработки создается пара из открытого и закрытого ключей, обеспечивающих безопасную загрузку. Закрытый ключ используется для создания подписи прошивки или данных, которая, в конечном счете, проверяются с помощью устройства DS28C36, установленной в конечном устройстве. Закрытый ключ никогда не покидает безопасную среду разработчика. Открытый ключ из этой пары сохраняется в регистре ключей Authority key микросхемы DS28C36 (изменяемый регистр DS28C36).
2. Закрытый ключ используется для вычисления цифровой подписи прошивки или данных.
3. Устройство DS28C36 с прошитым в ней открытым ключом соединено с главным процессором.
4. При запуске прошивка сначала извлекается менеджером загрузки процессора и передается в DS28C36 последовательными 64-байт блоками для вычисления хэша SHA-256.

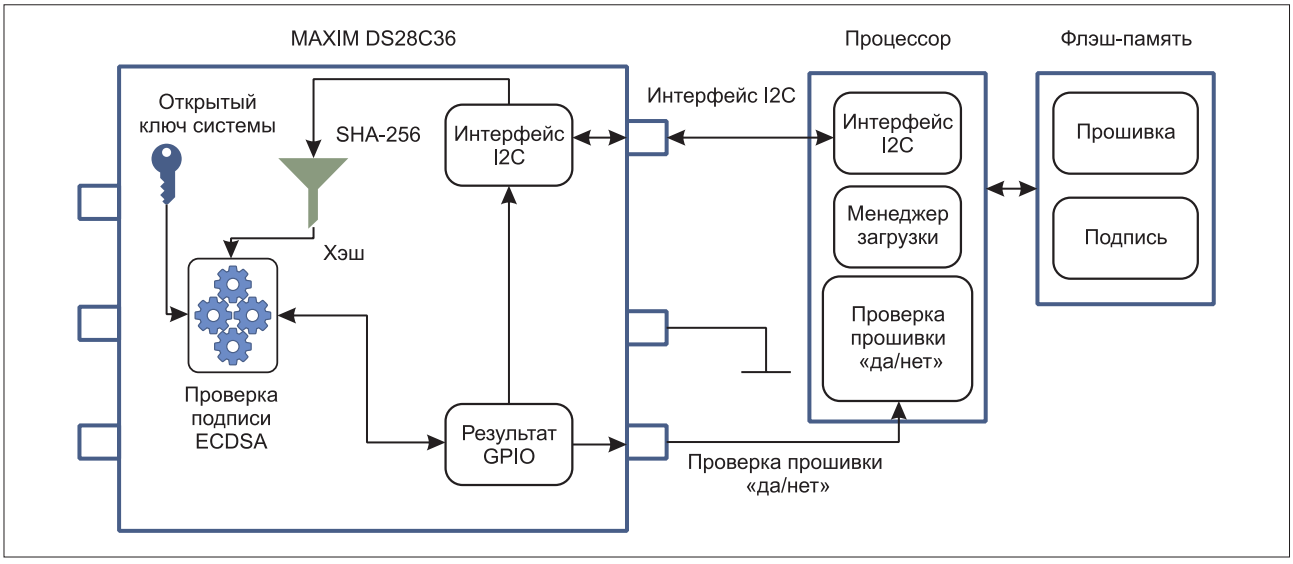

Рис. 5. **Безопасная загрузка при использовании DS28C36**

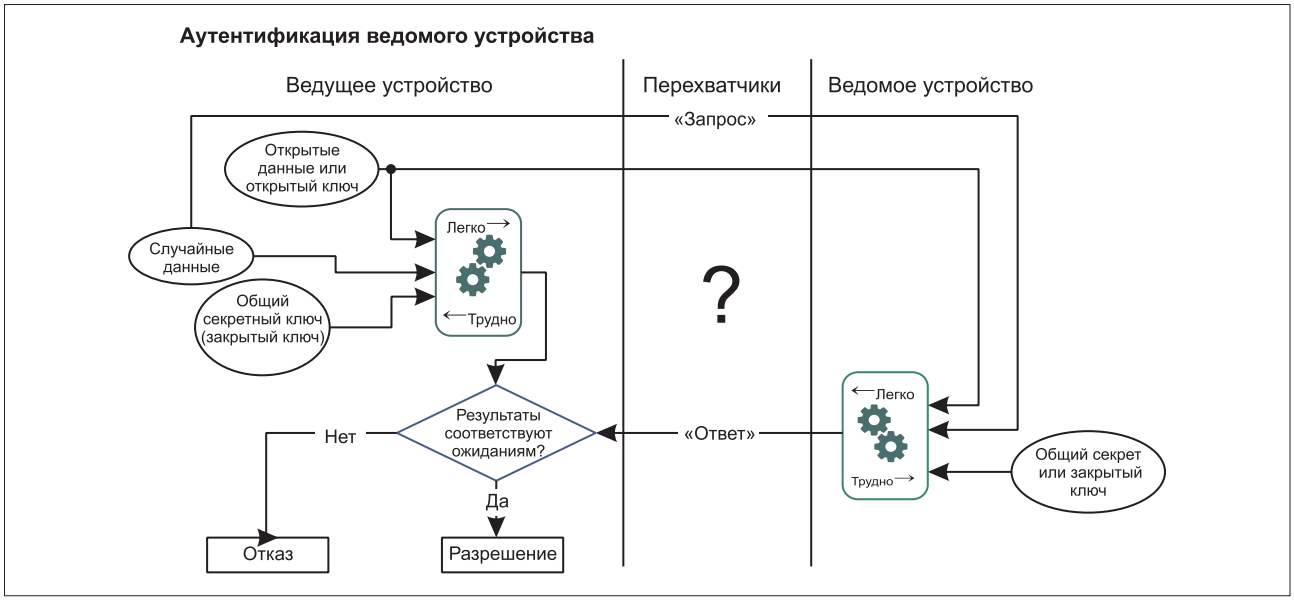

Рис. 6. **Аутентификация ведомого устройства в системе «ведущий–ведомый»**

5. После того как DS28C36 завершит вычисление хэша SHA-256, процессор передает подпись ECDSA, которая была сделана разработчиком устройства и добавлена к файлу.
6. После того как DS28C36 получило подпись ECDSA, процессор для выполнения проверки подписи отправляет команды на использование предустановленного системного открытого ключа.
7. Если DS28C36 подтверждает подпись, в процессор отправляется байт с результатом pass, а на выводе GPIO устанавливается логический 0. Таким образом процессор получает разрешение на запуск проверенной прошивки или на обновление данных.
8. Кроме того, для DS28C36 имеется опциональный механизм подписи ECDSA для проверки подлинности, выполняемой центром управления.

Таким образом, DS28C36 обеспечивает безопасную загрузку, устраняя угрозы для IoT-устройств. Предлагаемое аппаратное решение избавляет от сложных вычислений, связанных с проверкой подлинности и целостности прошивки или обновлений.

Для получения дополнительной информации об аппаратных системах безопасной загрузки компании Maxim предлагаем ознакомиться с защищенными устройствами для аутентификации.

Примеры обеспечения безопасности с помощью инструмента Security Lab см. на [3].

ДВУНАПРАВЛЕННАЯ АУТЕНТИФИКАЦИЯ ДЛЯ ЗАЩИТЫ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

Двунаправленная (или взаимная) аутентификация является важной частью безопасной коммуникации. Каждая сторона канала связи должна быть уверена, что ее партнеру можно доверять. Это обеспечивается путем доказательства владения закрытой информацией, которая передается между участниками или сохраняться в полном секрете, пока есть возможность доказать владение ею.

Системы симметричной аутентификации требуют обмена информацией между всеми участниками коммуникации. Эту информацию обычно называют «секретом». Секрет – это часть информации, не являющаяся общеизвестной; она известна только тем, кому это требуется. Секрет применяется совместно с симметричным алгоритмом аутентификации, например с SHA наряду с другими данными, совместно используемыми участниками. Способность генерировать совпадающую подпись обоими участниками коммуникации доказывает обладание секретом.

Системы асимметричной аутентификации (например, ECDSA) используют скрытую информацию, которая не передается между сторонами («закрытый ключ»), но предназначена для получения информации, являющейся общеизвестной («открытый ключ»). Правильное использование открытого ключа доказывает обладание закрытым ключом, поскольку для расшифровки сообщения, зашифрованного с помощью открытого ключа, необходимо иметь закрытый ключ.

АУТЕНТИФИКАЦИЯ ВЕДОМОГО УСТРОЙСТВА

Для аутентификации ведомого устройства в конфигурации «ведущий–ведомый» ему отправляются случайные данные («запрос»). Наряду с любыми данными, совместно используемыми устройствами, запрос выполняется через операцию подписи с секретным или закрытым ключом для получения «ответной» подписи. Эту подпись можно проверить ведущим устройством, поскольку оно владеет общим секретным ключом (открытым ключом), соответствующим закрытому ключу ведомого устройства. Общий ход этого процесса показан на рисунке 6.

Обычно аутентификация зависит от алгоритмов, создающих подписи, которые подтверждают обладание скрытой информацией участника, но не позволяют узнать содержание этой информации. Их называют необратимыми функциями. SHA и ECDSA являются примерами таких алгоритмов.

АУТЕНТИФИКАЦИЯ ВЕДУЩЕГО УСТРОЙСТВА

Чтобы доказать, что всем сторонам можно доверять, ведущее устройство также должно доказать ведомому устройству свою подлинность. Пример этого процесса показан на рисунке 7.

На рисунке 7 ведущее устройство записывает новые данные в ведомое устройство. Однако для завершения записи ведомое устройство должно проверить подлинность информации, потребовав от ведущего устройства создать подпись, основанную на этой информации и скрытых данных ведущего устройства (секрета или закрытого ключа). Используя либо общий секрет, либо открытый ключ, соответствующий закрытому ключу ведущего устройства, ведомое устройство проверяет подлинность подписи.

Использование необратимых функций позволяет любым перехватчикам видеть все передаваемые данные,

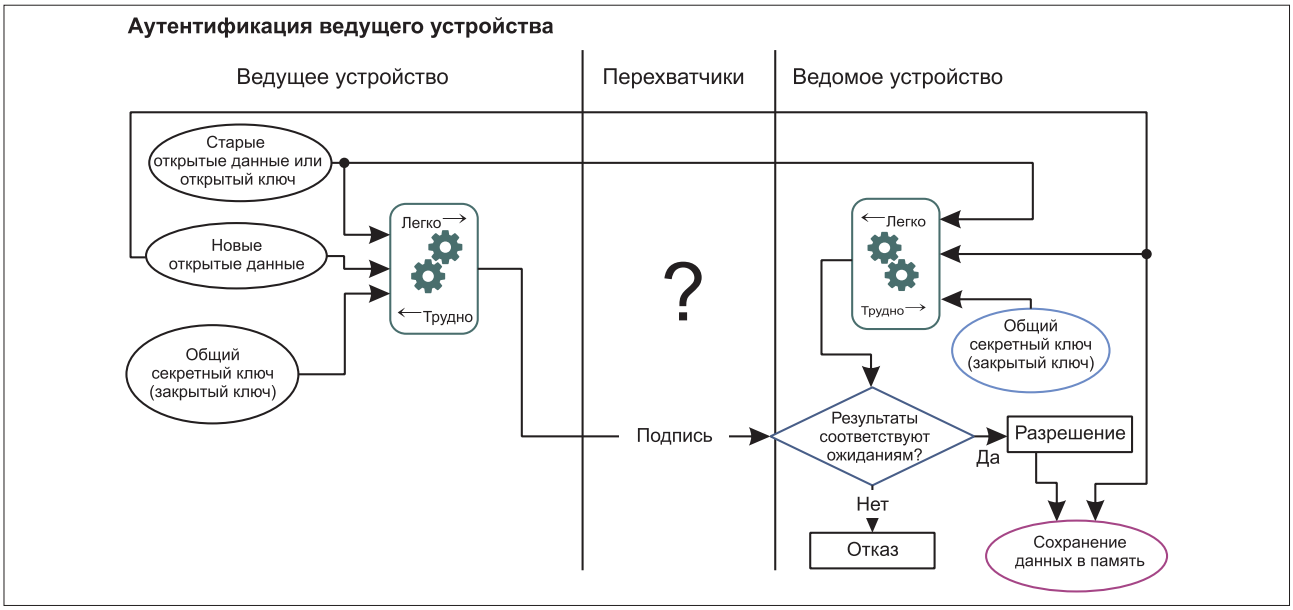

Рис. 7. **Ведущее устройство записывает новые данные в ведомое устройство**

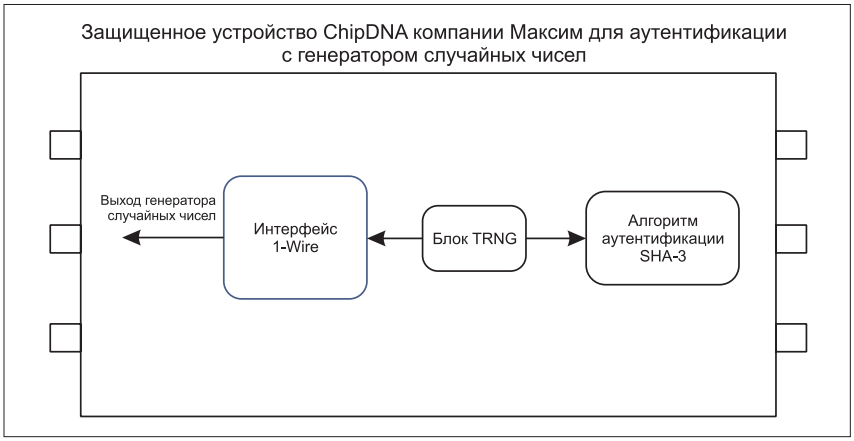

Рис. 8. **В защищенное устройство для аутентификации ChipDNA входит встроенный генератор истинно случайных чисел**

но не дает возможности определять скрытую информацию, с помощью которой была создана подпись. Без этой скрытой информации перехватчики не могут осуществить подмену.

Эта модель двусторонней аутентификации легко используется для обеспечения надежной защиты хранящейся в устройстве интеллектуальной собственности от фальсификаторов.

ВЫХОД TRNG И ТИПИЧНОЕ ИСПОЛЬЗОВАНИЕ

Защищенные устройства для аутентификации ChipDNA компании Maxim имеют встроенный генератор истинно случайных чисел (TRNG) (см. рис. 8), который необходим устройству для внутренних задач. Кроме этого существует команда, выдающая данные TRNG по запросу пользователя. В настоящее время максимальная длина выхода TRNG составляет 64 байт. Этот аппаратный источник случайных чисел, соответствующий требованиям NIST, может использоваться для криптографии, например для генерации «запроса» (случайного кода) хост-процессором.

Требования NIST/FIPS

Требования к генераторам истинно случайных чисел установлены в трех публикациях:

- NIST SP 800–90A;
- NIST SP 800–90B;
- NIST SP 800–90C.

Подробнее см. [4].

ЛИТЕРАТУРА

1. Защищенное устройство DS28C36 для аутентификации DeepCover с интерфейсом I2C//www.maximintegrated.com.

2. Защищенное устройство DS28E36 для аутентификации DeepCover с интерфейсом 1-Wire//www.maximintegrated.com.

3. Примеры обеспечения безопасности с помощью инструмента Security Lab//www.maxim-security.com.

4. Сайт NIST//www.nist.gov.