Аппаратно защищенные микросхемы семейства CryptoAuthentication: потенциальные применения ATECCx08A

Игорь КРИВЧЕНКО
ik@efo.ru

Шифрование содержит сообщение в секрете — только авторизованный получатель может его прочитать. Аутентификация гарантирует идентичность отправителя и целостность данных. Аутентификация и шифрование могут использоваться совместно для достижения более высокого уровня информационной безопасности.

ECDH (Elliptic Curve Diffie-Hellman) — алгоритм Диффи-Хеллмана на эллиптических кривых. Основывается на процедуре «запрос – ответ» с анонимным протоколом получения общего секрета (например, ключа шифрования). Это позволяет двум сторонам, каждая из которых имеет свою пару ключей «открытый – закрытый», сформировать общий секрет путем передачи несекретных данных по открытым каналам связи.

ECDSA (Elliptic Curve Digital Signature Algorithm) — алгоритм цифровой подписи на базе эллиптических кривых. Содержит две отдельные процедуры: для вычисления цифровой подписи и для ее проверки. Каждая процедура представляет собой последовательность определенных арифметических операций. Алгоритм вычисления цифровой подписи использует закрытый ключ, а алгоритм ее проверки — открытый ключ.

Вычисление/проверка цифровой подписи — процедура аутентификации, предусматривающая создание дайджеста данных, который затем зашифровывается закрытым ключом отправителя. Результатом является цифровая подпись. Для проверки цифровой подписи получатель самостоятельно хеширует полученные исходные данные, расшифровывает цифровую подпись открытым ключом отправителя, извлекая присланный дайджест, и сравнивает его с вычисленным. Если имеет место совпадение, то цифровая подпись считается верной.

Аппаратные криптографические блоки на кристалле ATECC508A совместно с безопасной областью памяти EEPROM для хранения ключей и сертификатов поддерживают полную 256-битную криптографию на эллиптических кривых, выполняют процедуры ECDSA и ECDH. Все это делает микросхему отличным выбором для защиты аксессуаров и сетевых узлов, включая решения IoT, поскольку нет необходимости в организации защищенного хранения секретных данных на стороне хоста. Интеграция микросхемы в систему проста благодаря широкому диапазону напряжений питания (2–5,5 В) и ультранизкому току потребления в энергосберегающем режиме (<150 нА). Ее отличительные особенности:

- ECDSA: FIPS186-3 алгоритм цифровой подписи на эллиптических кривых;
- ECDH: FIPS SP800-56A алгоритм Диффи-Хеллмана на эллиптических кривых;
- поддержка эллиптических кривых P256 стандарта NIST с длиной ключа 256 бит;
- хеш-алгоритм SHA-256 с опцией HMAC;
- уникальный 72-битный сериальный номер микросхемы;
- встроенный высококачественный генератор случайных чисел, одобренный FIPS;
- 10 кбит защищенной памяти EEPROM для ключей, сертификатов и данных;
- функционал для работы с однократно записываемой информацией и для учета числа использований;
- корпус SOIC8, UDFN8, XDFN8 и миниатюрный корпус с тремя выводами.

Микросхема ATECC108A является предшественницей ATECC508A. Основное отличие заключается в расширенном наборе поддерживаемых эллиптических кривых (поля F_p 256 и F_2^m 283), но при этом у ATECC108A отсутствует аппаратный шифратор ECDH. Хотя микросхема ATECC108A не рекомендована Atmel для новых разработок, ее серийное производство продолжается.

Микросхема ATECC508A хранит ключи, секреты и сертификаты на кристалле в защищенном от внешних атак аппаратном окружении, которое считается наиболее надежным с точки зрения защиты хранимой ин-

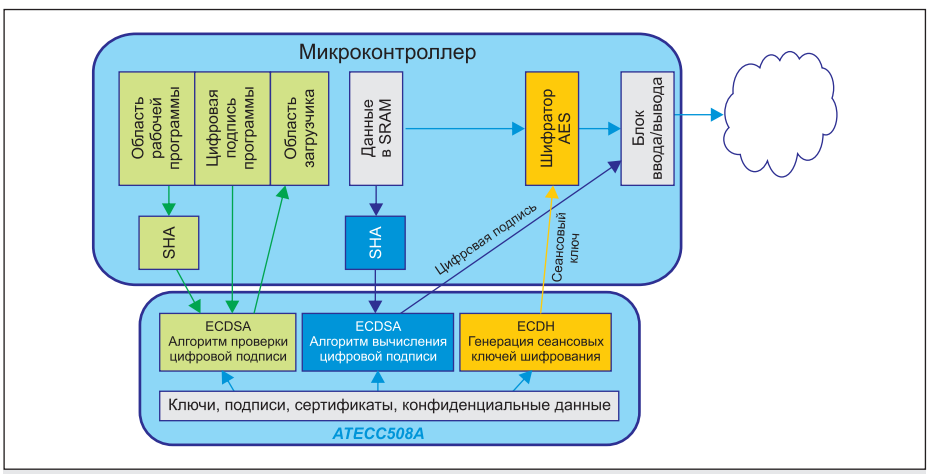

Рис. 1. Взаимодействие микросхемы ATECC508A и хост-микроконтроллера

формации. ATECC508A является отличным дополнением для любого микроконтроллера (рис. 1), способного выполнять симметричное шифрование (например, AES), благодаря встроенному алгоритму ECDH. Это обеспечивает хост-микроконтроллер надежными сеансовыми ключами шифрования путем выполнения несложных операций вызова макрокоманд из небольшого набора API. Микросхема ATECC508A реализует защищенную загрузку микропрограммы на этапах начальной загрузки или обновления (в том числе и удаленного), используя алгоритм проверки цифровой подписи ECDSA (зеленые блоки на рис. 1). В системах, где предусмотрена процедура генерации ключей по алгоритму ECDH с последующим шифрованием по алгоритму AES, целостность данных может быть обеспечена применением механизмов AES-CCM и AES-GCM. Следовательно, все три составляющие информационной безопасности поддерживаются микросхемой ATECC508A, а именно: конфиденциальность, целостность и аутентичность данных.

Полезной опцией микросхем семейства CryptoAuthentication является гибкий функционал учета числа использований (или обращений) на базе защищенных монотонных счетчиков. Основное применение данной функции — подсчет или ограничение количества использований аксессуара или конечного изделия, например картриджа принтера или аккумулятора смартфона. Другое применение — так называемое парное ограничение (pairing restriction), что предполагает точное совпадение пары «один хост — один определенный клиент», которой разрешена работа в системе. Это усиливает информационную безопасность: если клиент уже работает с каким-то определенным хостом, он не может быть использован другим хостом. Текущее значение счетчика также может быть связано с криптографическим ключом путем хеширования. Поскольку счетчик является монотонным, то есть работает только в одном направлении и никогда не сбрасывается, вычисляемый дайджест (MAC) всегда будет разным. Если рассматривать пример чернильного картриджа, это означает, что картридж не может быть вновь заправлен чернилами для повторного использования, поскольку не пройдет процедуру аутентификации из-за изменившегося MAC. Счетчик в микросхеме ATECC508A способен считать до 2 млн обращений, при этом может быть задействовано до 15 криптографических ключей.

Асимметричная аутентификация и сертификаты

В асимметричной криптографии, или криптографии с открытым ключом, реализованы не только механизмы шифрования, но и средства идентификации/аутентификации пользователей и конечных устройств. По сравнению с симметричной криптографией на первое место здесь вместо конфиденциальности выходит задача сохранения целостности и владения открытым ключом, то есть безопасного распределения открытых ключей.

Для того чтобы пользоваться всеми преимуществами асимметричной криптографии, участники процесса должны предоставить друг другу свои открытые ключи. Но поскольку открытый ключ можно подменить в процессе передачи, очень важно удостоверить пользователей в том, что, во-первых, ключ подлинный и, во-вторых, он был получен именно от той стороны, от которой предполагалось. Для безопасного распространения открытых ключей между большим количеством пользователей был разработан специальный механизм, в основе которого лежат сертификаты открытых ключей. Сертификаты помогают создать и поддерживать расширяемый, унифицированный и легко управляемый подход к распространению открытых ключей.

Сертификат открытого ключа свидетельствует о его связи с конечным пользователем и представляет собой защищенный от постороннего вмешательства набор данных. Чтобы подтвердить такую связь, одна или несколько третьих сторон, называемых удостоверяющими центрами (Certificate Authority, или CA), должны поручиться за идентичность пользователя. Для этого удостоверяющий центр создает сертификат, который содержит имя пользователя, его открытый ключ, идентификационную информацию и другие данные, и подписывает его своей уникальной цифровой подписью. После этого сертификат считается законченным и может свободно распространяться.

Представим, что вы — производитель системы распределенного контроля и управления. Гарантировать потребителю, что купленное им конечное изделие сделано именно на вашем предприятии, может сертификат этого изделия. Такой документ вы можете создать сами. Но вот гарантировать потребителю, что ваша фирма реально существует, должен кто-то другой — удостоверяющий центр. Именно CA может сделать для вас другой сертификат — уже вашего предприятия.

Для упрощенного описания процесса асимметричной аутентификации с помощью ECDSA применительно к микросхемам семейства CryptoAuthentication его можно разбить на два последовательных этапа. На первом этапе верифицируется открытый ключ клиента. Это осуществляет алгоритм проверки цифровой подписи на стороне хоста, извлекая необходимую информацию из переданного клиентом сертификата (или из цепочки сертификатов). Только в случае успешного результата проверки процесс аутентификации продолжается. На втором этапе верифицируется уже закрытый ключ клиента. Для этого хост посылает запрос (случайное число) клиенту, который тот подписывает своим закрытым ключом. Сформированная цифровая подпись пересылается обратно хосту, который запускает алгоритм ее проверки. В случае успеха клиент считается подлинным.

Отличительные характеристики:

- нет необходимости обновлять хост конфиденциальными данными «в поле» (открытый ключ может быть обновлен в любое время);
- обеспечивается повышенный уровень информационной безопасности, поскольку не требуется защищенное хранение закрытого ключа на стороне хоста (только на стороне клиента);
- микросхемы ATECC508A/108A имеют аппаратный блок ECDSA, что делает их очень легкими и удобными для применения.

Асимметричная аутентификация с использованием ECDSA: создание сертификатов

ECDSA «начинается» с сертификата. Посмотрим, как устроены сертификаты, а также каким образом они формируются

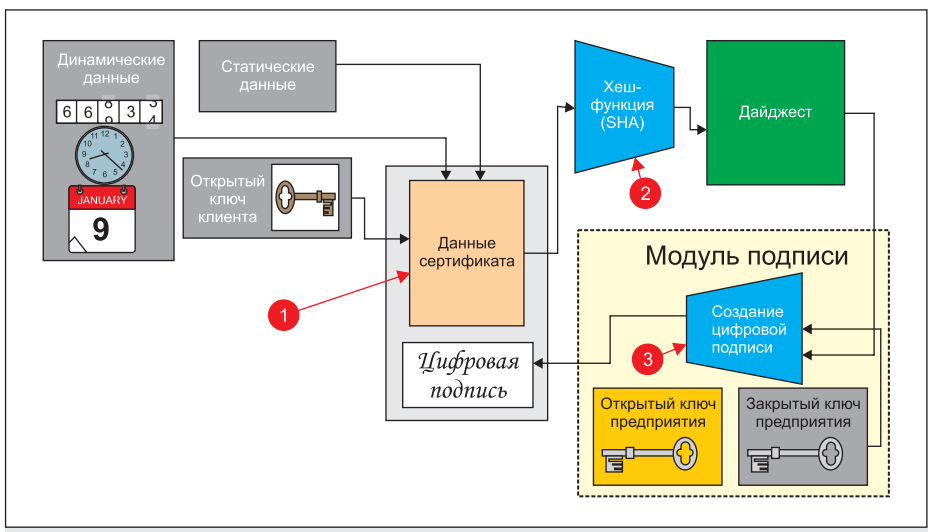

Рис. 2. Создание сертификата клиента

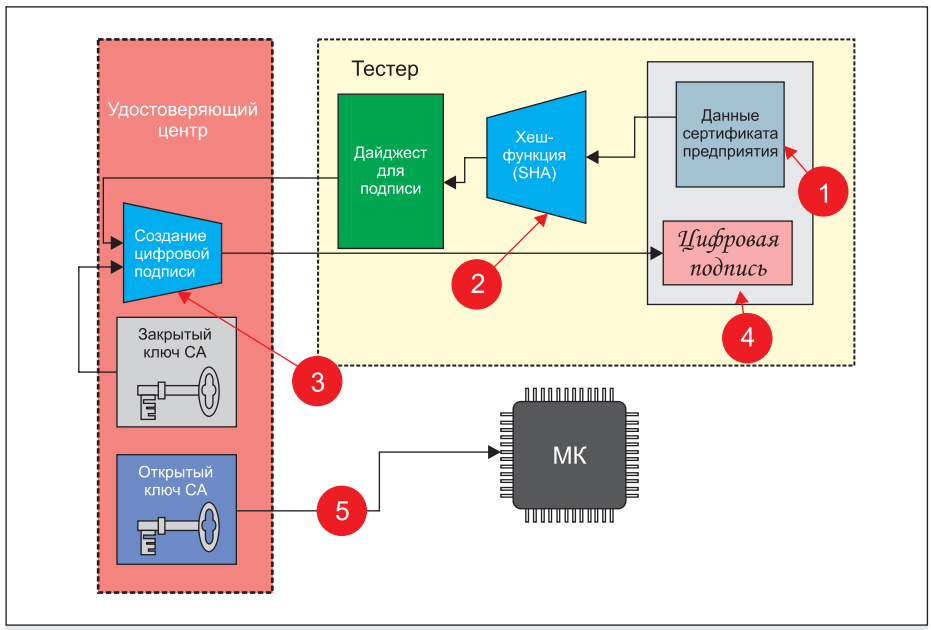

Рис. 3. Создание сертификата подписывающего

и загружаются в криптографическую микросхему. Это происходит на предприятии, выпускающем конечные изделия (далее — предприятие).

Сертификат клиента состоит из двух компонентов: данных сертификата и цифровой подписи. Процесс его создания, показанный на рис. 2, начинается в специальном программаторе (тестере), персонализирующем криптографическую микросхему. В тестер передаются данные будущего сертификата. Затем к тестеру присоединяется защищенное внешнее оборудование (модуль подписи), где безопасно хранится уникальный ключ предприятия. Задача этого модуля — создание цифровой подписи предприятия, которая затем объединяется в тестере с данными сертификата клиента.

Данные сертификата состоят из трех блоков: 1) статические данные; 2) динамические данные; 3) открытый ключ. Статические данные можно рассматривать, например, как часть базовой информации об изделии или о выпускающей его компании (наименование и состав изделия, имя и адрес компании и т. д.). Динамические данные представляют собой информацию, обычно изменяемую для каждой программируемой микросхемы или для группы микросхем. Такими данными могут быть серийный номер изделия или партии изделий, дата и время программирования, срок годности и т. п. Открытый ключ клиента — это третья составляющая данных сертификата. На шаге 1 в тестере все три составляющие объединяются, формируя данные сертификата. Напомним, что в асимметричной аутентификации открытый и закрытый ключи всегда формируются парами. Другими словами, если мы имеем открытый ключ, то где-то для этого же конечного изделия, содержащего криптографическую микросхему, должен существовать и закрытый ключ. Он надежно хранится в секрете и программируется в защищенную область памяти криптографической микросхемы на предприятии при производстве конечных изделий.

Вторая половина сертификата клиента — цифровая подпись предприятия. При формировании сертификата его данные используются двумя способами: в открытом, незашифрованном виде как составляющая часть сертификата и в виде цифровой подписи, которая получается (шаг 2) хешированием этих же исходных данных с последующим пропусканием полученного дайджеста в модуле подписи предприятия через алгоритм вычисления цифровой подписи (шаг 3). При этом используется хранящийся в модуле подписи закрытый ключ предприятия. Сформированная цифровая подпись пересылается в тестер, где объединяется с оригинальными (то есть не хешированными) данными, завершая тем самым процесс создания сертификата клиента (конечного изделия). Готовый сертификат теперь может загружаться в криптографическую микросхему. Алгоритм ECDSA P-256, используемый в микросхемах CryptoAuthentication, определяет 32-байтную длину дайджеста и алгоритм хеширования SHA-256.

В реальной жизни для подтверждения аутентичности самого предприятия, выпускающего конечные изделия, тоже может потребоваться его сертификат (еще называемый сертификатом подписывающего). Он выпускается одним из удостоверяющих центров.

Процедура создания сертификата предприятия аналогична процедуре создания сертификата клиента. Она показана на рис. 3. Сначала статические и динамические данные предприятия, необходимые для формирования сертификата в соответствии с общепринятыми стандартами, а также открытый ключ предприятия поступают в тестер. На шаге 1 в тестере все три составляющие объединяются, формируя данные сертификата. Затем те же данные хешируются в тестере (шаг 2), полученный дайджест пересылается в удостоверяющий центр, где он подписывается закрытым ключом удостоверяющего центра (шаг 3). Полученная цифровая подпись пересылается обратно на предприятие, где в тестере объединяется с данными сертификата предприятия (шаг 4), формируя тем самым законченный сертификат. Открытый ключ удостоверяющего центра также пересылается на предприятие (шаг 5) и программируется в управляющий микроконтроллер системы

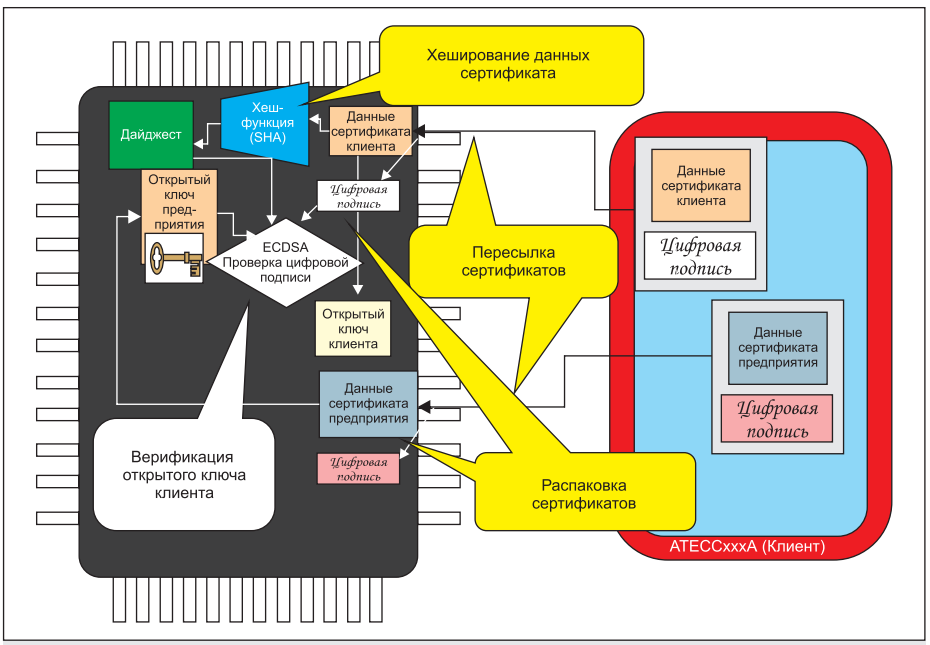

Рис. 4. Верификация открытого ключа клиента

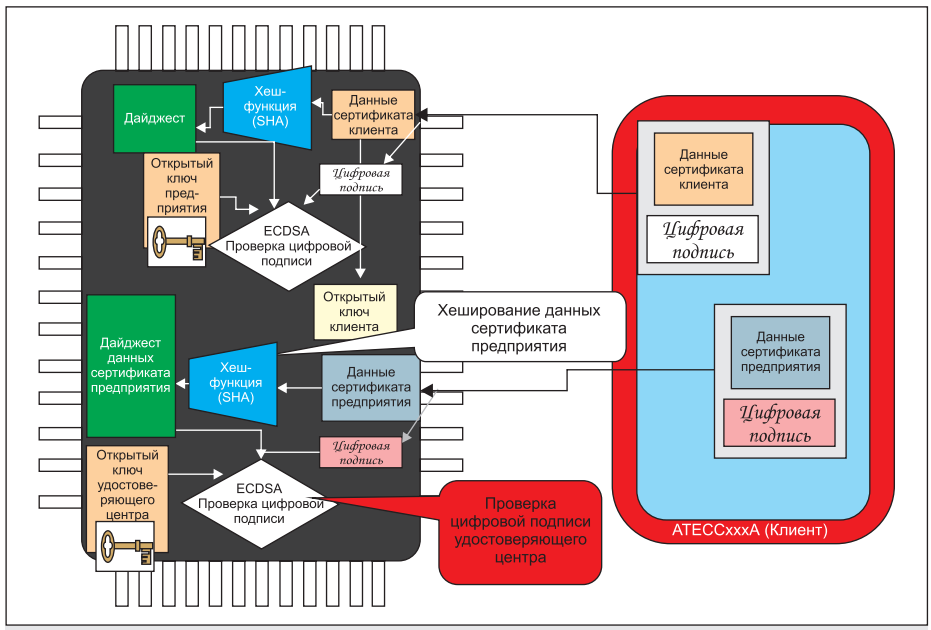

Рис. 5. Верификация открытого ключа предприятия

(хост). Это необходимо для последующего применения в реальной жизни при аутентификации.

И наконец, оба готовых сертификата — изделия и предприятия — загружаются с помощью тестера в защищенную энергонезависимую память программируемого криптографического устройства.

Асимметричная аутентификация: верификация открытого ключа клиента и открытого ключа предприятия

На этом этапе хост запрашивает от клиента информацию об его аутентичности, которая находится в составе сертификатов клиента и предприятия (рис. 4). Получив оба документа, хост извлекает из сертификата клиента его данные и открытый ключ, а также цифровую подпись, изготовленную в модуле подписи на предприятии. Хост также извлекает из сертификата предприятия его данные (в том числе открытый ключ) и подпись удостоверяющего центра. Затем хост отдельно хеширует наборы данных из сертификатов, создавая два дайджеста — клиента и предприятия. Извлеченный из сертификата предприятия открытый ключ подается на вход алгоритма проверки цифровой подписи ECDSA совместно с дайджестом данных сертификата клиента и цифровой подписью предприятия. Задача этих вычислений — верифицировать открытый ключ клиента.

В случае успеха хост переходит к следующему шагу — проверке аутентичности предприятия, то есть того, кто пока предположительно является производителем изделия (рис. 5). Без такой проверки открытый ключ клиента не может считаться подлинным. Это осуществляется путем верификации подписи удостоверяющего центра, которая приходит в составе сертификата предприятия. Цифровая подпись удостоверяющего центра подается на вход второго цикла алгоритма верификации цифровой подписи ECDSA вместе с дайджестом данных сертификата предприятия (вычислялся на предыдущем шаге) и открытым ключом удостоверяющего центра, который был заранее загружен в память микроконтроллера хоста.

Если оба последовательных цикла верификации ECDSA выполняются без ошибок, тогда открытый ключ клиента считается подлинным и верифицированным на всем пути — от изделия (через предприятие) до удостоверяющего центра.

Асимметричная аутентификация: верификация закрытого ключа

После успешного прохождения первого этапа вычислений ECDSA начинается второй этап — верификация закрытого ключа клиента. Напомним: конечной целью всего процесса асимметричной аутентификации является математическое доказательство того, что открытый и закрытый ключи клиента действительно составляют реальную пару ключей.

Хост генерирует запрос в виде случайного числа и посылает его клиенту (рис. 6). Криптоакселератор ECDSA в микросхеме ATECC508A/108A на стороне клиента генерирует новую цифровую подпись, используя это случайное число и закрытый ключ клиента. Полученная подпись пересылается обратно в хост. Теперь на стороне хоста случайное число, цифровая подпись и открытый ключ клиента, верифицированный на первом этапе, подаются в качестве входных данных на алгоритм проверки цифровой подписи ECDSA. Если проверка проходит успешно, хост может считать, что конечное устройство действительно содержит корректную пару ключей «открытый – закрытый». То есть что клиент — реальный.

Аппаратная и математическая реализация процесса асимметричной аутентификации достаточно сложна, но при использовании микросхем CryptoAuthentication это не имеет значения, поскольку Atmel обеспечивает пользователю легкую возможность реализовать криптографию без необходимости становиться экспертом в этой области. Применяется аппаратный шифратор ECDSA — проверенного и надежного алгоритма аутентификации. Используются преимущества криптографии на эллиптических

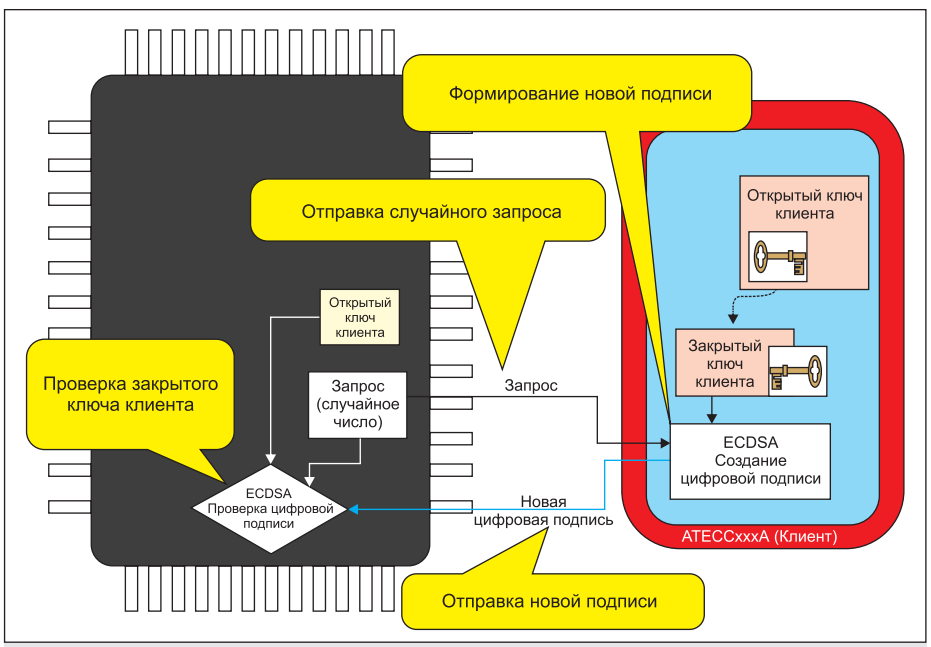

Рис. 6. Верификация закрытого ключа клиента

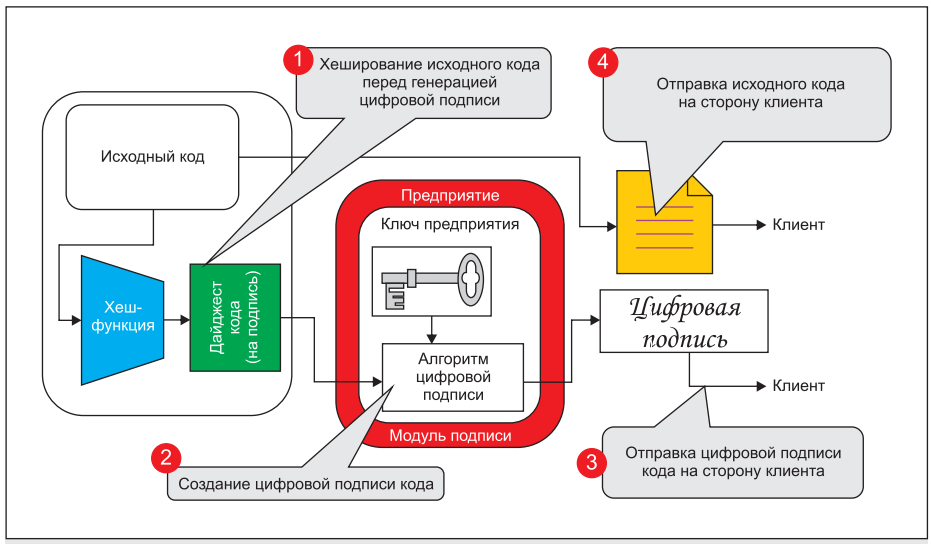

Рис. 7. Защищенная загрузка обновляемого программного обеспечения. Этап 1

кривых (высокая степень защиты, небольшая длина ключа, скорость работы). Как можно заметить, микросхема CryptoAuthentication не только хранит в своей защищенной памяти закрытый ключ, но и выполняет большой объем вычислений. Это значительно разгружает управляющие микроконтроллеры на сторонах хоста и клиента, а также не требует от них безопасного хранения ключа.

Асимметричная аутентификация: потенциальные применения

Аутентификация полезна для многих задач, например для защищенного хранения информации путем шифрования сохраняемых данных с индивидуальным секретным ключом для каждого файла. Другим приложением может быть аутентификация регистрируемых данных, скажем, серийного номера производителя, логов данных, конфигурационных параметров, калибровочных коэффициентов и т. п. Устройства CryptoAuthentication идеальны при реализации обмена секретными сеансовыми ключами потому, что закрытый (или секретный) ключ никогда не покидает криптографическую микросхему, позволяя микроконтроллеру хоста зашифровывать поток данных с помощью сеансового ключа. Они помогают организовать один из наиболее защищенных способов управления паролями, потому что все сравнения осуществляются внутри защищенного от атак кристалла и атакующие не могут найти ожидаемое значение. Помимо этого, пароли могут быть отображены на высокоэнтропийные ключи. И так далее.

Рассмотрим, к примеру, реализацию часто встречающейся задачи по защищенной загрузке обновляемого программного обеспечения. На первом этапе (рис. 7) микропрограмма подписывается в тестере на предприятии с помощью модуля подписи. Этот процесс напоминает создание сертификата клиента, только теперь подписываются не данные стандартного сертификата, а загружаемое в конечную систему ПО. Обновляемый образ кода (версия микропрограммы) в исходном, незашифрованном виде пересылается на сторону клиента. Одновременно пересылается и сгенерированная цифровая подпись этого образа кода. Это не сертификат, но очень похоже: есть данные, предназначенные для использования, — версия микропрограммы, и есть элемент аутентификации — уникальная цифровая подпись. Полученный на стороне клиента образ кода и соответствующая ему цифровая подпись загружаются в системную память управляющего микроконтроллера.

На втором этапе клиент проверяет аутентичность полученной микропрограммы — то, что она не была изменена или подменена в процессе передачи. С помощью микросхемы ATECC508A/108A для только что полученного образа кода создается цифровая подпись (подобно тому, как это делалось на предприятии), которая сравнивается с подписью, полученной вместе с микропрограммой со стороны хоста. В случае их совпадения считается, что полученный образ кода не был изменен или подменен в процессе передачи (рис. 8) и что он может загружаться в конечное устройство.

Еще один распространенный пример — задача одновременного обеспечения конфиденциальности, целостности и аутентичности при работе распределенной системы контроля и управления, в которой невозможно или нежелательно защищенным образом хранить секретный ключ на стороне хоста. Представим, что вам нужно со смартфона проконтролировать работу удаленного устройства — термостата, и во время сеанса связи термостат должен послать вам некоторый набор данных. По каким-то причинам вы хотите, чтобы обмен данными и командами между вашим смартфоном и термостатом был зашифрован. Рассмотрим вариант решения на базе микросхем ATECC508A/108A семейства CryptoAuthentication по схеме «фиксированный запрос — ответ» с промежуточным ключом и без хранения секретного ключа на стороне хоста.

Конфиденциальность данных при передаче обеспечивается их шифрованием (в нашем примере — алгоритмом симметричного шифрования AES). Обе стороны, обменивающиеся данными, должны при этом иметь один и тот же секретный ключ шифрования. Хорошим и надежным решением здесь является создание нового сеансового ключа для очередной сессии обмена данными, что существенно повышает уровень информационной безопасности в системе.

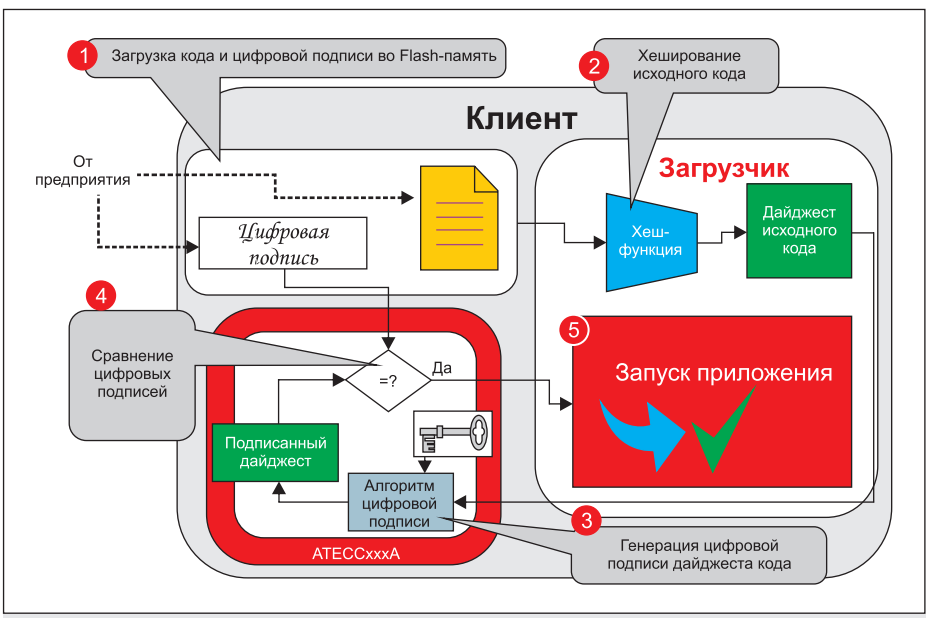

Рис. 8. Защищенная загрузка обновляемого программного обеспечения. Этап 2

В нашем случае, когда на стороне хоста ключи шифрования не хранятся защищенным образом, наиболее приемлем метод с фиксированным запросом. В тестере на предприятии каждый фиксированный запрос (некоторое случайное число) заранее хешируется с базовым секретным ключом, создавая при этом дайджест — промежуточный ключ шифрования. Таких фиксированных запросов может быть много. Полученные пары математически связанных чисел — запросов и соответствующих им ответов — вносятся в алгоритм ПО и записываются в энергонезависимую память управляющего микроконтроллера на стороне хоста. На предприятии во все клиентские микросхемы ATECC508A/108A также программируется базовый секретный ключ. Назовем его ключом конфиденциальности.

Процесс (рис. 9) начинается с фиксированного запроса, который хост пересылает клиенту (шаг 1). Этот запрос хешируется в криптографической микросхеме на стороне клиента с ключом конфиденциальности (шаг 2). Полученный дайджест (промежуточный ключ шифрования) затем вновь хешируется, но уже со случайным числом (например, комбинацией текущих значений даты и времени), генерируемым хостом и посылаемым клиенту (шаг 3). При этом формируется еще один дайджест, являющийся искомым уникальным сеансовым ключом шифрования (шаг 4). Теперь этот сеансовый ключ может быть подан на вход алгоритма AES в управляющем микроконтроллере клиента для зашифровки оригинального сообщения от термостата, которое затем посылается в хост (шаг 5). Хост хеширует свой промежуточный ключ шифрования, соответствующий посланному им ранее фиксированному запросу, с тем же случайным числом. При этом формируется идентичный сеансовый ключ (шаг 6) для расшифровки сообщения клиента (шаг 7).

Но перед дальнейшим применением полученных от термостата данных хосту нужно еще проверить их целостность, а также аутентичность их отправителя. Для проверки целостности (рис. 10) в схеме с фиксированным запросом рекомендуется использовать уже другие пары чисел, аналогично схеме конфиденциальности. Так и делается, но применяется второй секретный ключ — назовем его ключом целостности.

Хост посылает фиксированный запрос (шаг 1), который хешируется на стороне клиента в криптографической микросхеме вместе с хранящимся там ключом целостности (шаг 2). Дайджест этого ключа и входного запроса становится промежуточным ключом целостности на стороне клиента. Такой же промежуточный ключ целостности уже имеется в ПО на стороне хоста.

Вновь возвратимся к термостату: его пакет данных хешируется с промежуточным ключом целостности для создания кода аутентификации сообщения MAC (шаг 3). Но перед

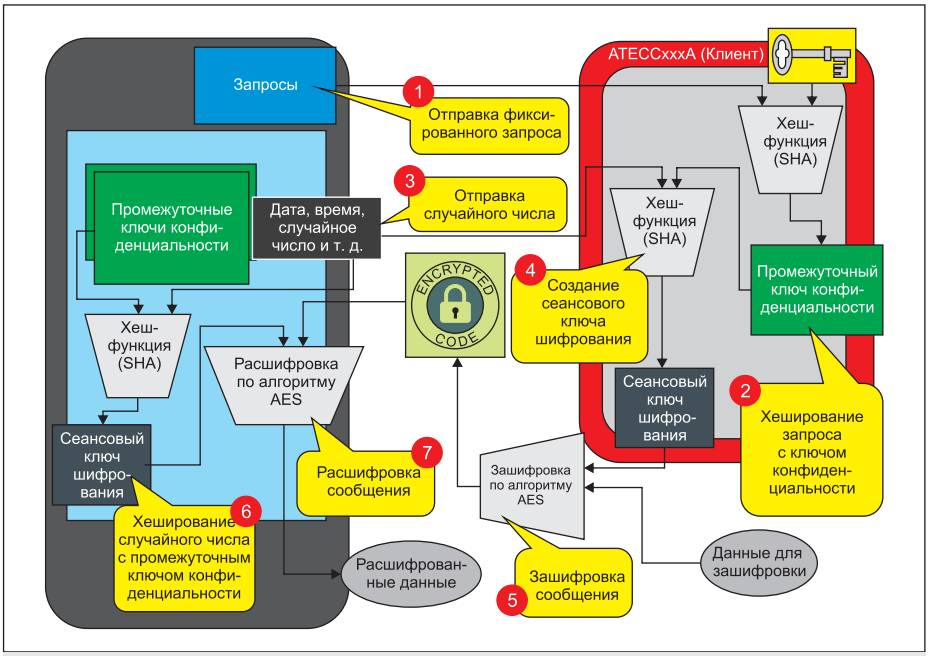

Рис. 9. Формирование сеансового ключа шифрования в системе с фиксированным запросом

этим данные должны быть подготовлены путем «подгонки» размера до 32 байт. Это осуществляется или путем превращения сообщения в дайджест, или путем добавления к сообщению дополнительных байтов, в зависимости от его исходного размера (32 байт — размер входных данных для алгоритма хеширования SHA-256). Созданный MAC затем добавляется к оригинальному сообщению, полученная комбинация зашифровывается сеансовым ключом и отправляется хосту (шаг 4).

Когда хост получает зашифрованное сообщение, он расшифровывает его, извлекая исходные данные и MAC. Для проверки данных на целостность хост хеширует их с ключом целостности таким же образом, как это делалось на стороне клиента (шаг 5). Вычисленный MAC сравнивается с расшифрованным MAC клиента (шаг 6). Если коды аутентификации совпадают, то целостность данных в сообщении считается подтвержденной.

Для завершения сеанса проводится аутентификация клиента — проверка того, что именно адресуемый термостат передал сообщение (рис. 11). Потребуется третий секретный ключ — аутентификации. С его помощью на стороне хоста заранее создается третий набор пар чисел «фиксированный запрос – ответ». Одновременно ключ аутентификации программируется на предприятии в криптографические клиентские микросхемы.

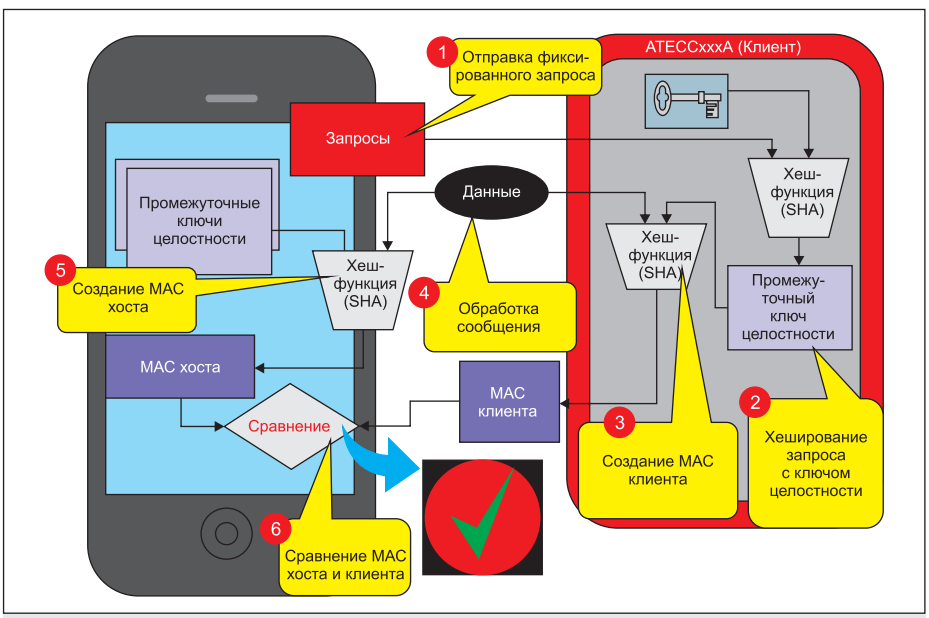

Рис. 10. Проверка целостности данных в системе с фиксированным запросом

Фиксированный запрос от хоста посылается клиенту (шаг 1) и хешируется там в криптографической микросхеме с ключом аутентификации (шаг 2), формируя промежуточный ключ аутентификации. Такой же промежуточный ключ аутентификации уже имеется в ПО на стороне хоста. Затем хост посылает клиенту новое случайное число (шаг 3). Клиент хеширует его с промежуточным ключом аутентификации для создания дайджеста (шаг 4), который пересылается в хост. На стороне хоста то же самое случайное число хешируется с промежуточным ключом, хранящимся в ПО хоста, для создания другого дайджеста (шаг 5). Полученный и вычисленный дайджесты сравниваются, и при их совпадении аутентичность клиента считается подтвержденной (шаг 6). Процесс завершен.

Поскольку микросхемы ATECC508A/108A обратно совместимы с ATSHA204A, то примеры потенциальных применений, представленных здесь и описанных в [2], могут быть реализованы с помощью любой микросхемы семейства CryptoAuthentication. Но все-таки главной особенностью ATECC508A/108A является встроенный алгоритм вычисления и проверки цифровой подписи ECDSA, а 508-я версия дополнительно имеет на кристалле аппаратный шифратор ECDH. Поэтому их рекомендуется применять для генерации сеансовых ключей шифрования и там, где для проверки данных и кода на целостность и аутентичность требуются цифровые подписи.

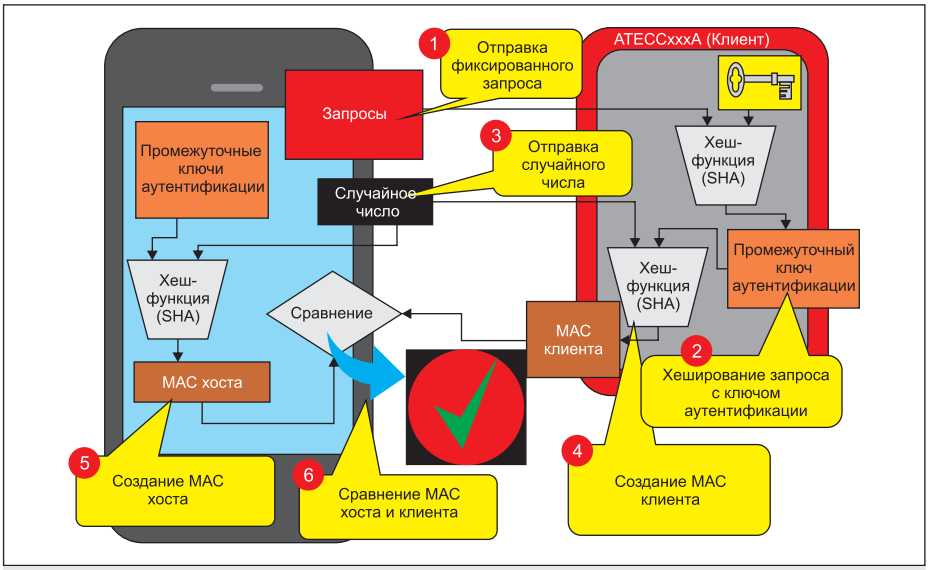

Рис. 11. Проверка аутентичности клиента в системе с фиксированным запросом

Конфиденциальность с использованием сеансовых ключей, созданных при помощи алгоритма ECDH

Конфиденциальность состоит в том, чтобы быть уверенным, что сообщение не может прочитать ни одна ненамеренная сторона. Обычно это достигается путем зашифровки и расшифровки сообщения на базе выбранного алгоритма шифрования, для чего обе стороны должны иметь одинаковый ключ, который будет являться входным параметром алгоритма шифрования. Отличный способ решения такой задачи распределения ключей — создание нового ключа для очередного сеанса связи путем обмена информацией, которую могут использовать только эти стороны. Тот факт, что новый сеансовый ключ формируется каждый раз для каждой новой сессии, существенно повышает уровень информационной безопасности. Данная процедура называется «согласование ключей», или «обмен ключами», и может быть выполнена с помощью широко распространенного и очень популярного алгоритма Диффи-Хеллмана (DH), в котором технология с открытым ключом используется для создания симметричного сеансового ключа.

Красивое и наглядное описание принципа работы алгоритма Диффи-Хеллмана можно найти в Интернете [3]. Главная особенность алгоритма состоит в том, что знание открытого ключа одной (или обеих) сторон не позволяет кому-либо еще заполучить или вычислить общий секретный сеансовый ключ шифрования — для этого нужен один из закрытых ключей. Практические реализации алгоритма Диффи-Хеллмана базируются или на множестве простых целых чисел (DH), или на множестве целых чисел, принадлежащих одной из криптографических эллиптических кривых (ECDH). При одинаковом уровне информационной безопасности операции ECDH выполняются значительно быстрее, чем операции DH, так как алгоритм работает с числами меньшей длины. В микросхеме ATECC508A реализован аппаратный шифратор ECDH.

На рис. 12 проиллюстрирована работа алгоритма ECDH для генерации общего ключа шифрования в системе, где на каждой из сторон, A и B, имеется микросхема ATECC508A. В этих криптографических устройствах безопасно хранятся пары ключей — открытый и закрытый. Для начала процесса стороны посылают друг другу свои открытые ключи (шаг 1). Затем полученные ключи отправителя подаются на вход алгоритма ECDH на каждой из сторон получателя вместе с еще двумя числами: заранее оговоренным между сторонами некоторым «базовым» значением и закрытым ключом получателя (шаг 2). В результате на каждой из сторон в защищенной аппаратной среде ATECC508A генерируется один и тот же сеансовый ключ шифрования. С помощью этого ключа сторона A зашифровывает свое исходное сообщение (шаг 3) по алгоритму AES и пересылает его стороне B, где оно расшифровывается по алгоритму AES с тем же сеансовым ключом (шаг 4).

Вернемся к рассмотренному ранее примеру с термостатом. Эту задачу можно решить иначе, применив на стороне клиента микросхему ATECC508A и другой набор крипто-

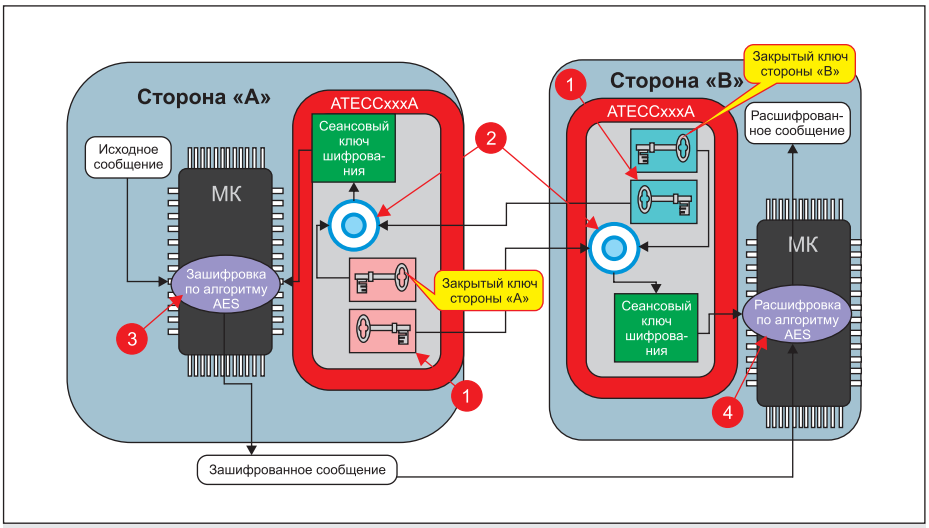

Рис. 12. Алгоритм ECDH

графических функций. Сначала проверим аутентичность термостата, используя алгоритм цифровой подписи ECDSA (рис. 13). Для этого в защищенную область памяти микросхемы ATECC508A, которая будет устанавливаться в наше удаленное устройство, на предприятии заранее программируются открытый и закрытый ключи клиента. Для начала процесса аутентификации термостат (по запросу хоста) посылает ему свой открытый ключ (шаг 1). Затем хост генерирует некоторое случайное или псевдослучайное число, представляющее собой запрос, и отправляет клиенту (шаг 2). Получив этот запрос, микросхема ATECC508A запускает процедуру создания цифровой подписи ECDSA, используя хранящийся в ней закрытый ключ клиента (шаг 3). Сформированная уникальная цифровая подпись отправляется хосту (шаг 4). Получив цифровую подпись, хост запускает у себя процедуру ее проверки (шаг 5). В качестве входных аргументов используются: проверяемая цифровая подпись, полученный ранее открытый ключ клиента и сгенерированное на шаге 2 случайное число. При успешной проверке аутентичность клиента считается подтвержденной.

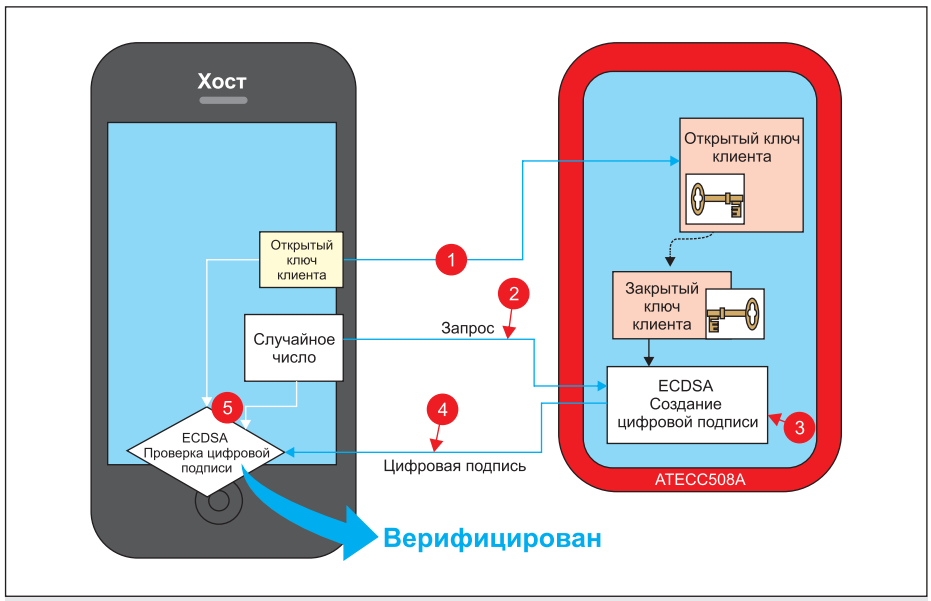

Рис. 13. Аутентификация клиента при помощи цифровой подписи ECDSA

Две следующие операции — шифрование и проверка целостности данных — выполняются за один этап. При этом используются все преимущества микросхемы ATECC508A. Сеансовый ключ шифрования генерируется с помощью уже упомянутого алгоритма ECDH, но этот же ключ будет использован и для проверки целостности данных, для чего он сначала хешируется в ATECC508A вместе с исходными незашифрованными данными. Полученный дайджест затем объединяется с этими исходными данными и зашифровывается текущим сеансовым ключом по алгоритму AES. Зашифрованный пакет отправляется на сторону хоста, где он обрабатывается аналогичным образом: исходные данные расшифровываются и одновременно проверяются на наличие возможных изменений в процессе передачи. Такая технология, использующая один и тот же ключ для шифрования и проверки целостности данных, называется «аутентифицированное шифрование» (authenticated encryption). Ее применение аналогично работе алгоритма шифрования AES в альтернативном режиме AES-CCM.

В итоге, применяя всего одну дополнительную микросхему ATECC508A, в конечной системе можно реализовать все три базовые составляющие информационной безопасности — конфиденциальность, целостность данных и аутентичность.

Заключение

Перед тем как выполнять задачи аутентификации в целевом приложении, микросхемы семейства CryptoAuthentication обязательно должны быть сконфигурированы определенным образом, запрограммированы и затем окончательно заблокированы. Этот процесс называется «персонализация». Он включает настройку желаемого профиля работы устройства с конфиденциальными данными (в том числе секретными ключами), которые должны быть защищены от постороннего вмешательства на каждом этапе процесса. Компания Atmel выпускает отладочные средства различного уровня сложности для ознакомления с принципами работы микросхем CryptoAuthentication, их программирования и персонализации. Об этих инструментах, а также о специальном сервисе компании Atmel по персонализации микросхем непосредственно на производстве Atmel в соответствии с требованиями конечного заказчика, планируется рассказать в следующей публикации.

Литература

1. www.atmel.com
2. Кривченко И. Аппаратно защищенные микросхемы семейства CryptoAuthentication: потенциальные применения ATSHA204A // Компоненты и технологии. 2015. № 10.
3. www.youtube.com/watch?v=3QnD2c4Xovk