DOI: 10.18721/JCSTCS.10403
УДК 004.021

ПРОТОКОЛ АУТЕНТИФИКАЦИИ ДЛЯ ПЕРЕДАЧИ ПРАВ ПОЛЬЗОВАНИЯ ЭЛЕКТРОННЫМИ УСТРОЙСТВАМИ

А.Я. Омётов[1], С.В. Беззатеев[2], Е.А. Кучерявый[3]

[1]Санкт-Петербургский государственный университет телекоммуникаций имени профессора М.А. Бонч-Бруевича, Санкт-Петербург, Российская Федерация;

[2] Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Российская Федерация;

[3] Национальный исследовательский университет «Высшая школа экономики», Москва, Российская Федерация

Количество носимой электроники в современном мире неуклонно растет. Традиционно за обеспечение безопасности электроники такого типа в условиях Интернета Вещей отвечали алгоритмы аутентификации. В статье представлен протокол аутентификации, специально разработанный для функционирования в условиях нестабильного соединения с удостоверяющим центром (центром сертификации). Подобные сценарии могут возникать и в случае повышенной нагрузки на существующую сеть, и в местах с низким сотовым покрытием или при функционировании в труднодоступных (удаленных) зонах. Рассмотрены ближайшие аналоги протокола, отмечены его преимущества относительного рассматриваемого сценария функционирования. Приведено детальное описание рабочих этапов протокола для обеспечения целостности и конфиденциальности данных владельца и пользователя устройства.

Ключевые слова: аутентификация; протокол; Интернет вещей; беспроводная связь; нестабильный канал.

Ссылка при цитировании: Омётов А.Я., Беззатеев С.В., Кучерявый Е.А. Протокол аутентификации для передачи прав пользования электронными устройствами // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2017. Т. 10. № 4. С. 29–40. DOI: 10.18721/JCSTCS.10403

AUTHENTICATING ELECTRONIC DEVICES FOR TEMPORARY USE: ACCESS RIGHTS MANAGEMENT PROTOCOL

A.Ya. Ometov[1], S.V. Bezzateev[2], Y.A. Koucheryavy [3]

[1] Bonch-Bruevich St. Petersburg State University of Telecommunications, St. Petersburg, Russian Federation;

[2] St. Petersburg National Research University of Information Technologies, Mechanics and Optics, St. Petersburg, Russian Federation;

[3] National Research University Higher School of Economics, Moscow, Russian Federation

The number of active wearable devices per user is growing daily. Conventionally, authentication algorithms were responsible for the security of such systems especially in terms of the Internet of Things (IoT) paradigm. This manuscript proposes an authentication protocol specifically designed for operation under unreliable connectivity to trusted authority constraints. Such situations may occur both in cases of network overload and in distant areas with low network coverage. The summary on existing solutions as well as the benefits brought by the solution are discussed. A detailed description of the protocol execution phases enabling integrity and privacy for the owner and the temporary user is also given.

Keywords: authentication; protocol; Internet of Things; wireless communication; unreliable connection.

Citation: Ometov A.Ya., Bezzateev S.V., Koucheryavy Y.A. Authenticating electronic devices for temporary use: access rights management protocol. St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems, 2017, Vol. 10, No. 4, Pp. 29–40. DOI: 10.18721/JCSTCS.10403

Введение

На протяжении развития сотовой связи и электронной торговли обеспечение безопасности при передаче данных является важнейшей задачей [1]. В настоящее время проверка подлинности пользователей, устройств Интернета вещей, событий и систем производится с помощью протоколов аутентификации [2]. Отличительная черта протоколов аутентификации, используемых в современных IP сетях, – необходимость наличия удостоверяющего центра (УЦ), который несёт ответственность также и за маршрутизацию сообщений между удалёнными узлами [3]. Основное достоинство таких систем – абсолютная доверительность УЦ, который берет на себя ответственность за генерацию, распределение, обновление и отзыв ключей/сертификатов всех пользователей системы [4].

В то же время развитие рынка носимой электроники диктует новые тренды в мире беспроводных технологий и информационной безопасности [5]. Однако сравнительно небольшие по размеру устройства, такие как носимая электроника и объекты умного дома, все ещё обладают невысокими вычислительными мощностями и ограничены в возможностях управления [6]. Рынок носимой электроники только начал развиваться, поэтому целевая аудитория пользователей ещё не достигла больших масштабов [7]. Общественность не готова приобретать новые дорогостоящие продукты носимой электроники. В связи с этим, для увеличения потребительской базы, недолговременная аренда таких устройств может послужить выгодным сценарием. С другой стороны, по стандартам современных производителей, смена пользователей может быть осуществлена только при условии полной очистки устройства до заводского состояния [8].

Данный сценарий – временная передача управления устройствами – может происходить как в местах с отличным сотовым покрытием (городах), так и в труднодоступных районах [9, 10] (горные курорты, морские круизы и т. п.). Во втором случае необходимость постоянного соединения с УЦ может оказаться серьёзной проблемой, что влечёт за собой трудности однозначного подтверждения владения устройством, обновления правил передачи, а также продления условий аренды.

Таким образом, задача данного исследования заключается в обеспечении возможности временной передачи управления устройствами в условиях непостоянного соединения с УЦ с целью предоставления гарантий целостности и конфиденциальности данных. Однако полный отказ от взаимодействия с УЦ и переход на полностью распределённые решения также является невозможным. Например, аренда устройств в контексте таких бизнес-моделей, как B2B или B2C, предполагает необходимость формального подтверждение факта передачи и возврата носимой электроники не только на нижнем уровне взаимодействия, но и на централизованном узле (сервере). Основная цель исследования – разработ-

ка семейства протоколов информационной безопасности, устраняющего указанную выше проблему.

Обзор ближайших аналогов и постановка задачи

К сожалению, современная научная литература не рассматривает проблему делегирования устройств в контексте нестабильного соединения с УЦ. Тем не менее далее приведён обзор работ, наиболее близких по проблематике.

В [11] представлен протокол, позволяющий снизить объёмы вычислений для устройств с существенно низким уровнем вычислительных ресурсов. В данном протоколе агрегирующий узел (АУ) отвечает за выпуск ключей дочерних устройств. Таким образом снижаются энергозатраты на устройствах с менее ёмкой батареей. Основным преимуществом предложенного решения является передача трудоёмкой процедуры рукопожатий генерации ключа на АУ. С другой стороны, недостатком является необходимость стабильного соединения с УЦ. Работа при наличии такого соединения требует прохождения минимум 7 и максимум 15 рукопожатий в фазе ассоциации.

В работе [12] представлен алгоритм, основанный на эллиптических кривых, который позволяет осуществлять аутентификацию устройств при помощи технологий связи беспроводных локальных сетей. Основными недостатками данной методики являются: необходимость постоянного наличия защищённого канала между аутентифицируемым устройством и УЦ, а также отсутствие возможности проверки истории взаимных аутентификаций.

Протокол, предлагаемый в данной статье, разработан с учётом проблем, описанных выше. Его основная задача – обеспечение аутентификации с целью последующего делегирования электронных устройств в обоих случаях – при наличии и отсутствии защищённого канала до УЦ. Благодаря возможности работы системы даже в случае отказа соединения с УЦ во время установления связи между владельцем устройства и потенциальным арендатором, протокол позволяет устранить недостатки известных технических решений. Система нацелена на обеспечение целостности данных и противодействие ряду атак на передаваемое устройство как со стороны владельца, так и со стороны арендатора устройства.

Описание протокола аутентификации

Предлагаемый протокол обеспечивает функционирование в сценариях с потенциально нестабильным соединением от АУ к УЦ, а также в случае наличия возможности осуществления прямых соединений между АУ и делегируемым устройством и АУ разных пользователей.

Поставленная задача решается за счёт комбинирования решений, рассматриваемых в системах с инфраструктурой открытых ключей, которые отвечают за начальную генерацию ключей и сертификатов, а также решений, позволяющих осуществлять прямое соединение при отсутствии связи с инфраструктурой открытых ключей.

Основные варианты криптографических решений для потенциального использования в данном протоколе предложены ниже. Для получения результата хеш-функции данных могут быть использованы алгоритмы ГОСТ Р 34.11-2012, SHA-2, SHA-3, BLAKE2 [13], для ЦУ возможно использование классических протоколов криптосистем с открытым ключом [14–16]. Защита от атак типа «человек посередине» (атака посредника) может осуществляться посредством классического протокола Diffie-Hellman [17] или Elliptic curve Diffie-Hellman (ECDH) [18].

Протокол аутентификации для делегации прав пользования электронными устройствами состоит из определённых операционных этапов (шагов): (1) начальная ассоциация нового устройства; (2) передача устройства на определённый промежуток времени; (3) возврат устройства в пользование владельцу; (4) диссоциация устройства.

Начальная ассоциация нового устройства. Исходными данными являются уникальный идентификатор пользователя ID (например, user@mail.com), который используется для взаимодействия между участниками протокола (УЦ, пользователями системы и

делегируемым устройством). Участниками протокола на этом этапе являются ранее не используемое делегируемое электронное устройство (ДЭУ) и агрегирующий узел (например, смартфон пользователя). Далее, УЦ выпускает сертификаты для АУ пользователя и нового ДЭУ, заверяя целостность прошивки ДЭУ и его однозначную принадлежность к АУ.

Шаги протокола данного этапа показаны на рис. 1. Здесь и далее владельцем ДЭУ считается пользователь Alice; арендатором устройства – пользователь Bob. Защищённым каналом определяется устойчивое к атаке «человек посередине» соединение. Каждый пользователь обладает уникальным идентификатором ID_A типа alice@mail.com, собственным секретным и публичным ключами SK_A и PK_A. УЦ хранит публичные ключи всех пользователей и соответствующие сертификаты $cert_A = sign_{cloud}(PK_A)$. Каждое ДЭУ имеет предустановленное заводское ПО и заводской ключ для сброса к начальным настройкам. Предполагается наличие безопасного таймера и защищённого хранилища, которые считаются доверенными. На защищённую часть устройства возлагаются следующие функции:

1) Безопасный таймер/часы – предоставление доверенного значения времени. Без данного свойства невозможно отслеживание основного параметра делегации – времени.

2) Защищённое хранилище – необходимо для хранения конфиденциальной информации, ключей и т. п.

3) Доверенный функционал устройства – функции, связанные напрямую с информационной безопасностью, такие как, например, хеш-функции, генераторы псевдослучайных последовательностей и т. п.

Также в процессе функционирования протокола вводится ключ S_A, ассоциированный с пользователем Alice. Данный ключ позволяет обеспечить симметричное шифрование передаваемых и хранимых на ДЭУ данных, а также служит дополнительной защитой от утечки конфиденциальной информации пользователя Alice, хранимой на отдельно взятом ДЭУ. Дальнейшие шаги протокола обозначены соответствующими цифрами на рис. 1.

1. Пользователь Alice генерирует секретный ключ S_A для нового ДЭУ w_i и передаёт его по защищённому каналу.

2. ДЭУ w_i передаёт результат исполнения хеш-функции от собственного программного обеспечения на УЦ с использованием защищённого канала $(hash(SW_i))$.

3. Пользователь Alice передаёт свой публичный ключ PK_A и идентификатор ID_A на УЦ.

4. УЦ генерирует сертификат УЦ

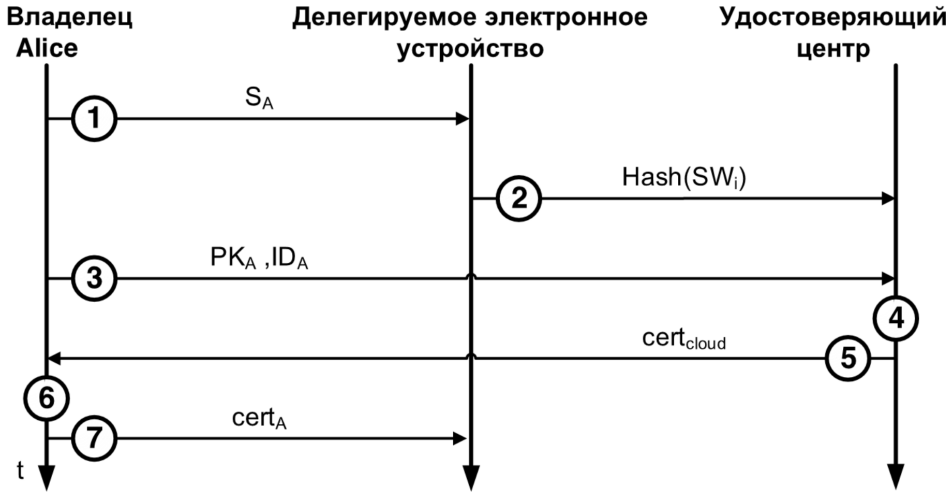

Рис. 1. Схема пошагового выполнения этапа ассоциации ДЭУ

Fig. 1. Association phase execution

для Alice. Сертификат имеет вид $cert_{cloud} = sign_{cloud}(w_i, ID_A, hash(SW_i))$. Данный шаг выполняется с целью обеспечения целостности системных данных.

5. УЦ, используя защищённый канал, передаёт $cert_{cloud}$ пользователю Alice.

6. Пользователь Alice подписывает сертификат, полученный от УЦ, своим секретным ключом $cert_A = sign_A(cert_{cloud})$.

7. Пользователь Alice передаёт сертификат на ДЭУ.

Передача устройства на некоторое время в пользование другому пользователю. Функционирование системы может происходить в двух режимах: при наличии соединения с УЦ и при отсутствии соединения. В первом случае за аутентификацию полностью отвечает УЦ, соответственно рис. 1. В случае отсутствия стабильного соединения с УЦ протокол делегирования требует наличия прямого соединения между АУ пользователей и делегируемым ДЭУ. Шаги протокола, выполняемые на данном этапе, показаны на рис. 2 и 3, протоколы 2А и 2Б, соответственно. Данный сценарий описывает передачу электронного устройства в аренду с указанием условий и времени пользования ДЭУ.

Протокол 2А – наличие стабильного соединения у обоих пользователей (рис. 2).

1. Пользователь Alice устанавливает максимальное время делегирования (t_d) для ДЭУ w_i, используя служебное сообщение, подписанное SK_A, как $m[D]_A = sign_A(w_i, t_d, ID_A, ID_B, \{прочие\ условия\ делегации\})$, согласно шагу 1.

2. Пользователь Alice передаёт $m[D]_A$ на УЦ, используя защищённый канал, согласно шагу 2.

3. УЦ проверяет подлинность сообщения от Alice, используя PK_A. В случае непрохождения проверки протокол прекращает работу, согласно шагу 3.

4. УЦ подписывает сообщение делегирования $m[D]_{cloud} = sign_{cloud}(m[D]_A)$, согласно шагу 4.

5. УЦ передаёт $m[D]_{cloud}$ и $cert_A$ для Bob, в соответствии с шагом 5.

6. Пользователь Alice передаёт на ДЭУ служебное сообщение $m[C(S_A)]_A$, удаляющее секретный ключ S_A с ДЭУ, в соответствии с шагом 6.

7. Если пользователь Bob **не доверяет** Alice, протокол выполняет шаг 7. ДЭУ сбрасывается к заводским установкам. Сброс происходит с сохранением $m[D]_{cloud}$ и сертификата $cert_{cloud}$ в защищённом хранилище. Данные значения получены на шаге 6 протокола инициализации с целью сохранения целостности данных и подтверждения права собственности при отсутствии соединения с УЦ. АУ пользователя Bob сравни-

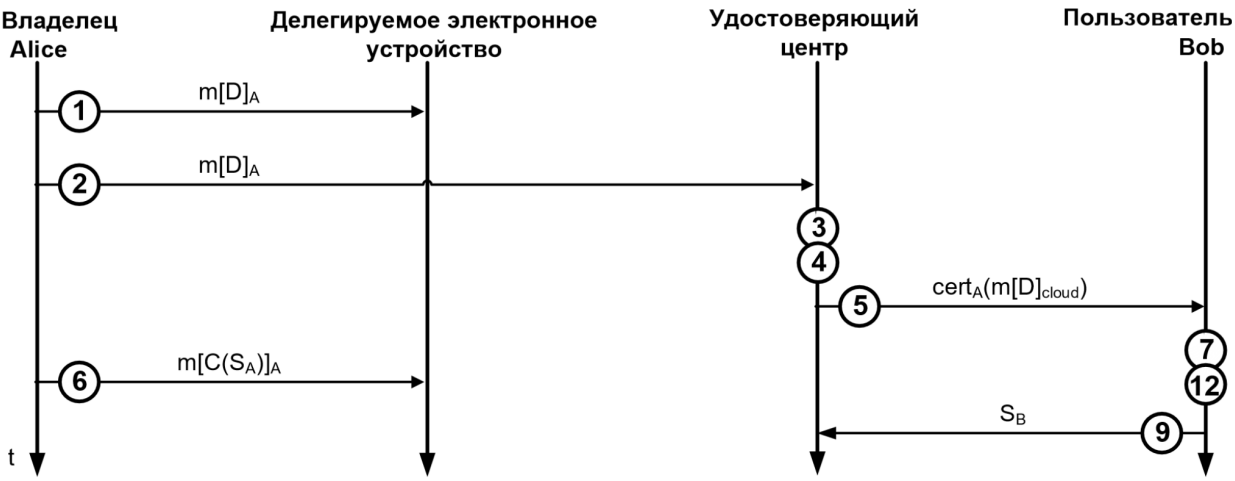

Рис. 2. Схема пошагового выполнения этапа делегирования ДЭУ при наличии стабильного соединения с УЦ

Fig. 2. Delegation phase execution: case of stable infrastructure connectivity

вает результат хеш-функции от текущего программного обеспечения ДЭУ с сохранённым в $cert_{cloud}$. Если они не совпадают, алгоритм прекращает исполнение. Таким образом пользователь Bob теряет возможность использовать устройство ДЭУ, т. к. предполагается, что ПО могло быть скомпрометировано владельцем. Важно отметить, что защищённые таймер и хранилище остаются неизменными даже при сбросе к заводским настройкам.

8. Если пользователь Bob **доверяет** Alice, программная компонента ДЭУ остается неизменной, и временный пользователь имеет возможность пользоваться программным обеспечением владельца устройства.

9. Пользователь Bob генерирует секретный ключ S_B для прямого взаимодействия с ДЭУ, согласно шагу 12 протокола.

10. Пользователь Bob передаёт S_B на ДЭУ по защищённому каналу, согласно шагу 9.

11. Для обеспечения целостности данных пользователь Bob рассчитывает новое значение $sign_B(w_i, SW_i)$.

12. В случае истечения таймера делегирования t_d на стороне ДЭУ, параметры устройства сбрасываются к заводским установкам с сохранением содержимого защищённого хранилища. Таймер может быть удалённо обновлён в случае одновременного наличия соединения с УЦ пользователя и владельца при использовании служебного сообщения $m[D]_A = sign_A(w_i, t_d, ID_A, ID_B,$ {*прочие условия делегации*}).

Протокол 2B – отсутствие стабильного соединения хотя бы у одного из пользователей (рис. 3).

1. Пользователь Alice устанавливает максимальное время делегирования t_d на ДЭУ w_i, используя служебное сообщение, подписанное SK_A, как $m[D]_A = sign_A(w_i, t_d, ID_A, ID_B,$ {*прочие условия делегации*}), согласно шагу 1.

2. Пользователь Alice передаёт $cert_A$ пользователю Bob по защищённому каналу, согласно шагу 2.

3. Bob проверяет подлинность $cert_A$, используя $cert_{cloud}$. В случае непрохождения проверки протокол прекращает работу, согласно шагу 3.

4. Пользователь Alice передаёт на ДЭУ служебное сообщение $m[C(S_A)]_A$, удаляющее секретный ключ S_A с ДЭУ, согласно шагу 4.

5. Если пользователь Bob **не доверяет** Alice, функционирование протокола выполняется в соответствии с шагом 5. ДЭУ сбрасывается к заводским установкам. Сброс происходит с сохранением $m[D]_{cloud}$ и сертификата $cert_{cloud}$ в защищённом хранилище. Данные значения были получены на шаге 6 протокола инициализации с целью сохранения целостности данных и под-

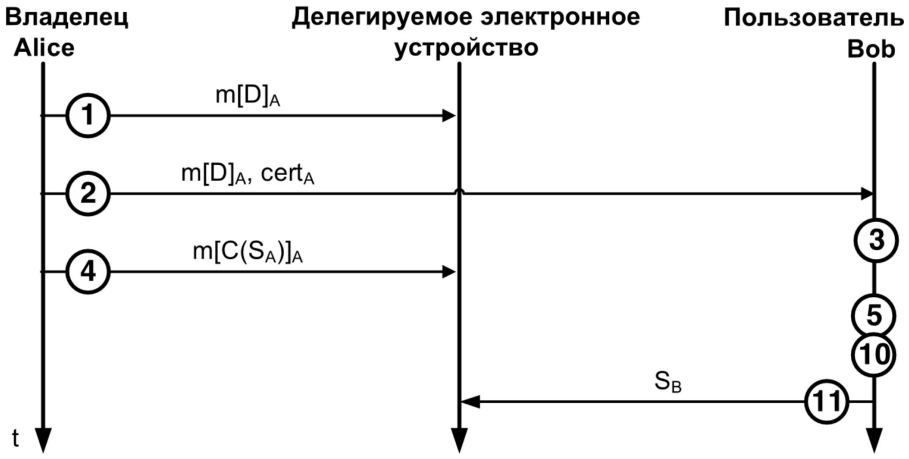

Рис. 3. Схема пошагового выполнения этапа делегирования ДЭУ при отсутствии стабильного соединения с УЦ

Fig. 3. Delegation phase execution: case of unreliable infrastructure connectivity

тверждения права собственности при отсутствии соединения с УЦ. АУ пользователя Bob сравнивает результат хеш-функции от текущего программного обеспечения ДЭУ с сохранённым в $cert_{cloud}$. Если они не совпадают, алгоритм прекращает исполнение. Таким образом, пользователь Bob теряет возможность использовать устройство ДЭУ, т. к. предполагается, что ПО могло быть скомпрометировано владельцем. Важно отметить, что защищённые таймер и хранилище остаются неизменными даже при сбросе к заводским настройкам.

6. Если пользователь Bob **доверяет** Alice, программная компонента ДЭУ остаётся неизменной, и временный пользователь имеет возможность пользоваться программным обеспечением владельца устройства.

7. Пользователь Bob генерирует секретный ключ S_B для прямого взаимодействия с ДЭУ, согласно шагу 10.

8. Пользователь Bob передаёт S_B на ДЭУ посредством защищённого канала, согласно шагу 11.

9. Для обеспечения целостности данных пользователь Bob рассчитывает новое значение $sign_B(w_i, SW_i)$.

10. В случае истечения времени делегирования t_d на стороне ДЭУ, параметры устройства сбрасываются к заводским установкам с сохранением содержимого защищённого хранилища. Таймер может быть обновлён с использованием служебного сообщения $m[D]_A = sign_A(w_i, t_d, ID_A, ID_B,$ {*прочие условия делегирования*}) при наличии прямого соединения между пользователями.

11. Пользователь Bob передаёт S_B на ДЭУ по защищённому каналу, согласно шагу 9 на рис. 4.

12. Для обеспечения целостности данных пользователь Bob рассчитывает новое значение $sign_B(w_i, SW_i)$.

Возврат устройства в пользование владельцу после временного использования ДЭУ другим пользователем. Функционирование системы может происходить в двух режимах: при наличии соединения с УЦ и без такого соединения. Соответствующие шаги показаны на рис. 4 (протокол 3А) и 5 (протокол 3Б). В первом случае, аналогично делегированию, за аутентификацию полностью отвечает УЦ. В случае отсутствия стабильного соединения с УЦ алгоритм возврата устройства требует наличия прямых соединений между АУ пользователей и делегируемым ДЭУ.

Протокол 3А – наличие стабильного соединения у обоих пользователей (рис. 4)

1. Пользователь Bob генерирует служебное сообщение, подписанное SK_B, как $m[R]_B = sign_A(w_i, R)$, согласно шагу 1.

2. Пользователь Bob передаёт $m[R]_B$ на

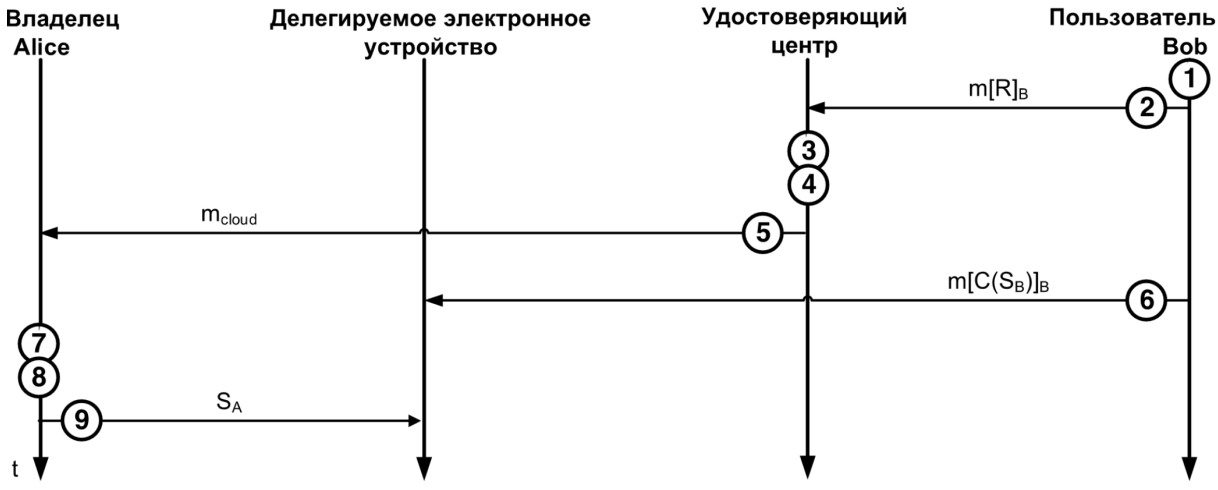

Рис. 4. Схема пошагового выполнения этапа возврата ДЭУ при наличии стабильного соединения с УЦ

Fig. 4. Reclaiming phase execution: case of stable infrastructure connectivity

УЦ, используя защищённый канал, согласно шагу 2.

3. УЦ проверяет подлинность сообщения от Bob с использованием PK_B. В случае непрохождения проверки протокол прекращает работу, согласно шагу 3.

4. УЦ подписывает сообщение возврата $m[R]_{cloud} = sign_{cloud}(m[R]_B)$, согласно шагу 4.

5. УЦ передаёт $m[R]_{cloud}$ пользователю Alice, согласно шагу 5.

6. Пользователь Bob передаёт на ДЭУ служебное сообщение $m[C(S_B)]_B$, удаляющее секретный ключ S_B с ДЭУ, согласно шагу 6.

7. Если пользователь Alice **не доверяет** Bob, выполняется шаг 7. ДЭУ сбрасывается к заводским установкам с сохранением данных в защищённом хранилище. АУ пользователя Alice сравнивает результат хеш-функции от текущего программного обеспечения ДЭУ с сохранённым в $cert_{cloud}$. Если они не совпадают, протокол прекращает свою работу. Таким образом, пользователь Alice теряет возможность использовать устройство ДЭУ, т. к. предполагается, что ПО могло быть скомпрометировано владельцем. Важно отметить, что защищённые таймер и хранилище остаются неизменными даже при сбросе к заводским настройкам.

8. Если пользователь Alice **доверяет** Bob, программная компонента ДЭУ остаётся неизменной, а владелец устройства имеет возможность пользоваться программным обеспечением, установленным в процессе делегирования.

9. Пользователь Alice генерирует секретный ключ S_A для прямого взаимодействия с ДЭУ, согласно шагу 8.

10. Пользователь Alice передаёт S_A на ДЭУ, используя защищённый канал, согласно шагу 9.

11. Для обеспечения целостности данных пользователь Alice рассчитывает новое значение $sign_A(w_i, SW_i)$. В защищённом хранилище ДЭУ находятся только $cert_A$ и $cert_{cloud}$.

Протокол 3B – отсутствие стабильного соединения хотя бы у одного из пользователей (рис. 5)

1. Пользователь Bob генерирует служебное сообщение, подписанное SK_B, как $m[R]_B = sign_B(w_i, R)$, согласно шагу 4.

2. Пользователь Bob передаёт $m[R]_B$ пользователю Alice по защищённому каналу, согласно шагу 2.

3. Alice проверяет подлинность $m[R]_B$, используя $cert_{cloud}$. В случае непрохождения проверки протокол прекращает свою работу, согласно шагу 3.

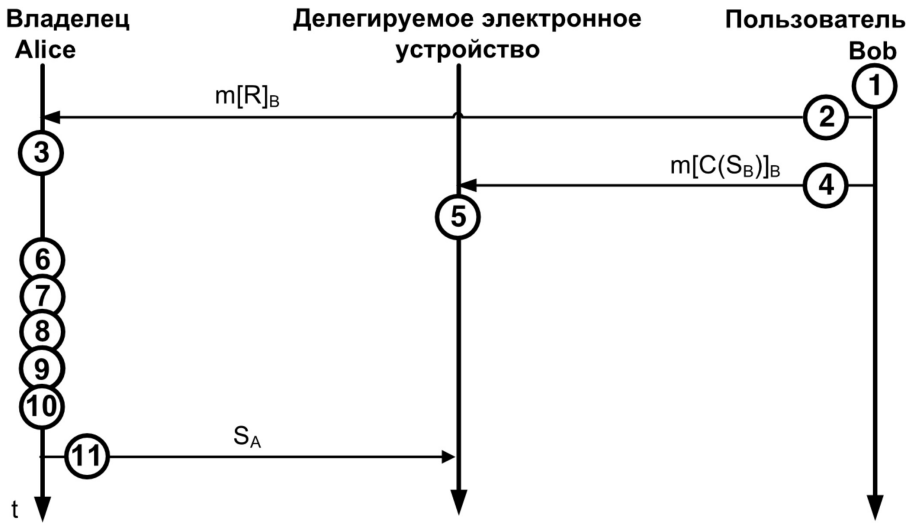

Рис. 5. Схема пошагового выполнения этапа возврата ДЭУ при отсутствии стабильного соединения с УЦ

Fig. 5. Reclaiming phase execution: case of unreliable infrastructure connectivity

4. Пользователь Bob передаёт на ДЭУ служебное сообщение $m[C(S_B)]_B$, удаляющее секретный ключ S_B с ДЭУ, согласно шагу 4.

5. Если пользователь Alice **не доверяет** Bob, функционирование протокола происходит согласно шагу 5. ДЭУ сбрасывается к заводским установкам с сохранением данных в защищённом хранилище. АУ пользователя Alice сравнивает результат хеш-функции от текущего программного обеспечения ДЭУ с сохранённым в $cert_{cloud}$. Если они не совпадают, протокол прекращает свою работу. Таким образом, пользователь Alice теряет возможность использовать устройство ДЭУ, т. к. предполагается, что ПО могло быть скомпрометировано владельцем. Важно отметить, что защищённые таймер и хранилище остаются неизменными даже при сбросе к заводским настройкам.

6. Если пользователь Alice **доверяет** Bob, программная компонента ДЭУ остаётся неизменной, а владелец имеет возможность пользоваться программным обеспечением временного пользователя устройства.

7. Пользователь Alice генерирует секретный ключ S_A для прямого взаимодействия с ДЭУ, согласно шагу 10.

8. Пользователь Alice передаёт S_A на ДЭУ по защищённому каналу, согласно шагу 11.

9. Для обеспечения целостности данных пользователь Alice рассчитывает новое значение $sign_A(w_i, SW_i)$. В защищённом хранилище ДЭУ находятся только $cert_A$ и $cert_{cloud}$.

Диссоциация устройства. В данном случае возможны два варианта функционирования протокола: ручная и автоматическая диссоциация (сброс настроек устройства к заводским). При ручной диссоциации владелец и ДЭУ связываются по прямому каналу. АУ владельца передаёт управляющее сообщение, сбрасывающее ДЭУ к состоянию заводских настроек. Автоматическая диссоциация возможна по предустановленному таймеру, который производит аналогичную процедуру по истечению времени делегирования устройства в случае его невозврата.

Протокол 4A – ручная диссоциация ДЭУ

1. Владелец Alice генерирует служебное сообщение, подписанное SK_A, как $m[F]_A = sign_A(w_i, F)$.

2. Пользователь Alice передаёт $m[R]_B$ на ДЭУ по защищённому каналу.

3. ДЭУ сбрасывается к заводским настройкам с очисткой защищённого хранилища.

4. Устройство ДЭУ может быть восстановлено только при использовании заводского ключа (например, PIN) и при наличии соединения с УЦ.

Протокол 4B – автоматическая диссоциация ДЭУ

Данный протокол выполняется в случае, когда устройство не было возвращено согласно условиям делегирования, утеряно или украдено. Все личные данные владельца и пользователя должны быть уничтожены с целью предотвращения потенциального вредоносного использования.

1. В процессе начальной ассоциации устройства владелец Alice опционально генерирует служебное сообщение, подписанное SK_A, как $m[E]_A = sign_A(w_i, t_e)$, где t_e является значением таймера обратного отсчёта автоматической диссоциации. ДЭУ обращается к таймеру в процессе эксплуатации с целью обнаружения потенциальной переработки.

2. В случае переработки ДЭУ сбрасывается к заводским настройкам, очищая защищённое хранилище.

3. Устройство ДЭУ может быть восстановлено только при использовании заводского ключа (например, PIN) и при наличии соединения с УЦ.

Возможные атаки на протокол

После детального рассмотрения предложенного протокола важно отметить потенциальные атаки на его работу. Одной из таких атак является фишинг. В данном случае злоумышленник Eve может представиться доверенным пользователем Bob с ID_B в процессе делегации между Alice и Bob. Если Alice не может проверить подлинность источника запроса, атака считается успешной. Основная область применения атаки

находится в сфере носимой электроники, т. к. устройства данного типа часто не обладают дополнительным визуальным каналом (например, дисплеем) для дополнительного подтверждения процедуры делегации. В современном мире не существует решения, полностью защищающего от фишинг атак, однако методики многофакторной аутентификации позволяют понизить вероятность успеха атаки.

Другой немаловажной атакой на протокол является классическая атака посредника. В нашем случае злоумышленник Eve запрашивает $m[D]_A$ для Bob (ID_B) от Alice, представляясь Alice (ID_A), и одновременно передаёт $m[D]_A$ для пользователя Bob. В результате Eve не получает возможности использования устройства, однако, может следить за процессом делегирования.

Одной из интереснейших атак является атака с использованием заражённого устройства. Рассмотрим следующий пример. После перехвата злоумышленником корректного $m[D]_A$ о делегировании устройства w_k, Eve создаёт заражённое устройство, которое представляется w_k и передаёт всегда корректное значение $hash(SW_k)$. Такое устройство позволяет, например, следить за активностью пользователя Bob.

В то же время протоколы, использующие цифровую подпись, особенно уязвимы с точки зрения конфиденциальности, поскольку они, как правило, вовлекают однозначное подтверждение со стороны доверенной сущности. Иными словами, пользователь не может позднее отречься от факта делегирования. Протоколы, основанные на доказательствах с нулевым разглашением, полагаются на это свойство для повышения безопасности, однако, вызывают дополнительные осложнения в области конфиденциальности.

Заключение

Представлен протокол аутентификации и контроля электронных устройств, разработанный для использования в условиях нестабильной связи с удостоверяющим центром. Алгоритмы, реализующие этапы функционирования предлагаемого протокола, могут быть реализованы как в виде программного обеспечения для универсального компьютера любой архитектуры, так и в виде аппаратного обеспечения для специализированного компьютера любой архитектуры. Протокол может использоваться в местах с отсутствием инфраструктуры, т. к. для реализации делегирования не требуется постоянного наличия соединения с УЦ.

Работа выполнена при поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере в рамках программы «УМНИК» по договору № 8268ГУ2015 от 02.12.2015 г.

СПИСОК ЛИТЕРАТУРЫ

1. **Kim H.J., Chang H.S., Suh J.J., Shon T.S.** A study on device security in IoT convergence // Proc. of Internat. Conf. on Industrial Engineering, Management Science and Application. IEEE, 2016. Pp. 1–4.

2. **Saadeh M., Sleit A., Qatawneh M., Almobaideen W.** August. Authentication techniques for the Internet of Things: A survey // Proc. of Cybersecurity and Cyberforensics Conference. IEEE, 2016. Pp. 28–34.

3. **Nix J.A.** M2M and IoT technologies, Llc. Systems and methods for “machine-to-machine” (M2M) communications between modules, servers, and an application using Public Key Infrastructure (PKI) // U.S. Patent 9,276,740, 2016.

4. **Sicari S., Rizzardi A., Grieco L.A., Coen-Porisini A.** Security, privacy and trust in Internet of Things: The road ahead // Computer Networks. 2015. No. 76. Pp.146–164.

5. **Andreev S., Galinina O., Pyattaev A., Gerasimenko M., Tirronen T., Torsner J., Sachs J., Dohler M., Koucheryavy Y.** Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap // IEEE Communications Magazine. 2015. Vol. 53. No. 9. Pp.32–40.

6. **Arias O., Wurm J., Hoang K., Jin Y.** Privacy and security in Internet of Things and wearable devices // IEEE Transactions on Multi-Scale Computing Systems. 2015. Vol. 1. No. 2. Pp. 99–109.

7. **Wei J.** How wearables intersect with the cloud and the Internet of Things: Considerations for the developers of wearables // IEEE Consumer Electronics Magazine. 2014. Vol. 3. No. 3. Pp. 53–56.

8. **Simon L., Anderson R.** Security analysis of android factory resets // Proc. of 4th Mobile Security Technologies Workshop. 2015.

9. **Abrardo A., Fodor G., Tola B.** Network coding schemes for device-to-device communications based relaying for cellular coverage extension // Proc. of 16th Internat. Workshop on Signal Processing Advances in Wireless Communications. IEEE, 2015. Pp. 670–674.

10. **Ometov A., Zhidanov K., Bezzateev S., Florea R., Andreev S., Koucheryavy Y.** Securing network-assisted direct communication: The case of unreliable cellular connectivity // Proc. of Trustcom/BigDataSE/ISPA. IEEE, 2015. Vol. 1. Pp. 826–833.

11. **Hummen R., Ziegeldorf J.H., Shafagh H., Raza S., Wehrle K.** Towards viable certificate-based authentication for the Internet of Things // Proc. of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy. ACM, 2013. Pp. 37–42.

12. **Lee Y.K., Batina L., Verbauwhede I.** Untraceable RFID authentication protocols: Revision of EC-RAC // Proc. of Internat. Conf. on RFID. IEEE, 2007. Pp. 178–185.

13. **Aumasson J.P., Neves S., Wilcox-O'Hearn Z., Winnerlein C.** BLAKE2: simpler, smaller, fast as MD5 // Proc. of Internat. Conf. on Applied Cryptography and Network Security. Springer, 2013. Pp. 119–135.

14. **Wander A.S., Gura N., Eberle H., Gupta V., Shantz S.C.** Energy analysis of public-key cryptography for wireless sensor networks // Proc. of 3rd IEEE Internat. Conf. on Pervasive Computing and Communications. IEEE, 2015. Pp. 324–328.

15. **Elgamal T.** A public key cryptosystem and a signature scheme based on discrete logarithms // IEEE Transactions on Information Theory. 1985. Vol. 31. No. 4. Pp.469–472.

16. **He D., Zeadally S.** An analysis of rfid authentication schemes for Internet of Things in healthcare environment using elliptic curve cryptography // IEEE Internet of Things Journal. 2015. Vol. 2. No. 1. Pp.72–83.

17. **Diffie W., Hellman M.** New directions in cryptography // IEEE Transactions on Information Theory. 1976. Vol. 22. No. 6. Pp. 644–654.

18. **Joux A.** A one round protocol for tripartite Diffie–Hellman // International Algorithmic Number Theory Symp. Springer, 2000. Pp. 385–393.

Статья поступила в редакцию 10.07.2017.

REFERENCES

1. **Kim H.J., Chang H.S., Suh J.J., Shon T.S.** A study on device security in IoT convergence. *Proceedings of International Conference on Industrial Engineering, Management Science and Application*, IEEE, 2016, Pp. 1–4.

2. **Saadeh M., Sleit A., Qatawneh M., Almobaideen W.** August. Authentication techniques for the Internet of Things: A survey. *Proceedings of Cybersecurity and Cyberforensics Conference*, IEEE, 2016, Pp. 28–34.

3. **Nix J.A.** M2M and IoT Technologies, Llc. Systems and methods for "machine-to-machine" (M2M) communications between modules, servers, and an application using Public Key Infrastructure (PKI). *U.S. Patent 9,276,740, 2016.*

4. **Sicari S., Rizzardi A., Grieco L.A., Coen-Porisini A.** Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, 2015, No. 76, Pp.146–164.

5. **Andreev S., Galinina O., Pyattaev A., Gerasimenko M., Tirronen T., Torsner J., Sachs J., Dohler M., Koucheryavy Y.** Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap. *IEEE Communications Magazine*, 2015, Vol. 53, No. 9, Pp. 32–40.

6. **Arias O., Wurm J., Hoang K., Jin Y.** Privacy and security in Internet of Things and wearable devices. *IEEE Transactions on Multi-Scale Computing Systems*, 2015, Vol. 1, No. 2, Pp. 99–109.

7. **Wei J.** How wearables intersect with the cloud and the Internet of Things: Considerations for the developers of wearables. *IEEE Consumer Electronics Magazine*, 2014, Vol. 3, No. 3, Pp.5 3–56.

8. **Simon L., Anderson R.** Security analysis of android factory resets. *Proceedings of 4th Mobile Security Technologies Workshop*, 2015.

9. **Abrardo A., Fodor G., Tola B.** Network coding schemes for device-to-device communications based relaying for cellular coverage extension. *Proceedings of 16th International Workshop on Signal Processing Advances in Wireless Communications*, IEEE, 2015, Pp. 670–674.

10. **Ometov A., Zhidanov K., Bezzateev S., Florea R., Andreev S., Koucheryavy Y.** Securing network-assisted direct communication: The case of unreliable cellular connectivity. *Proceedings of Trustcom/BigDataSE/ISPA*, IEEE, 2015, Vol. 1, Pp. 826–833.

11. **Hummen R., Ziegeldorf J.H., Shafagh H., Raza S., Wehrle K.** Towards viable certificate-based authentication for the Internet of Things.

Proceedings of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy, ACM, 2013, Pp. 37–42.

12. **Lee Y.K., Batina L., Verbauwhede I.** Untraceable RFID authentication protocols: Revision of EC-RAC. *Proceedings of International Conference on RFID*, IEEE, 2007, Pp. 178–185.

13. **Aumasson J.P., Neves S., Wilcox-O'Hearn Z., Winnerlein C.** BLAKE2: simpler, smaller, fast as MD5. *Proceedings of International Conference on Applied Cryptography and Network Security*, Springer, 2013, Pp. 119–135.

14. **Wander A.S., Gura N., Eberle H., Gupta V., Shantz S.C.** Energy analysis of public-key cryptography for wireless sensor networks. *Proceedings of 3rd IEEE International Conference on Pervasive Computing and Communications*, IEEE, 2015, Pp. 324–328.

15. **Elgamal T.** A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 1985, Vol. 31, No. 4, Pp. 469–472.

16. **He D., Zeadally S.** An analysis of rfid authentication schemes for Internet of Things in healthcare environment using elliptic curve cryptography. *IEEE Internet of Things Journal*, 2015, Vol. 2, No. 1, Pp.72–83.

17. **Diffie W., Hellman M.** New directions in cryptography. *IEEE Transactions on Information Theory*, 1976, Vol. 22, No. 6, Pp. 644–654.

18. **Joux A.** A one round protocol for tripartite Diffie–Hellman. *International Algorithmic Number Theory Symp.,* Springer, 2000, Pp. 385–393.

Received 10.07.2017.

СВЕДЕНИЯ ОБ АВТОРАХ / THE AUTHORS

ОМЁТОВ Александр Ярославич
OMETOV Aleksander Ya.
E-mail: alexander.ometov@gmail.com

БЕЗЗАТЕЕВ Сергей Валентинович
BEZZATEEV Sergey V.
E-mail: bsv@aanet.ru

КУЧЕРЯВЫЙ Евгений Андреевич
KOUCHERYAVY Yevgeni A.
E-mail: ykoucheryavy@hse.ru

© Санкт-Петербургский политехнический университет Петра Великого, 2017