DEVELOPMENT OF AUTHENTICATION MODEL FOR CLOUD COMPUTING

Vishniakov A.S.[1], Makarov A.E.[2], Utkin A.V.[3], Zazhogin S.D.[4], Bobrov A.V.[5] (Russian Federation) Email: Vishniakov560@scientifictext.ru

[1]Vishniakov Alexandr Sergeevich – Lead System Engineer,
SYSTEM INTEGRATOR «KRASTCOM»;
[2]Makarov Anatoly Evgenevich – Solutions Architect,
ROSTELECOM INFORMATION TECHNOLOGY,
MOSCOW;
[3]Utkin Alexander Vladimirovich – Senior Engineer,
INTERNATIONAL SYSTEM INTEGRATOR EPAM SYSTEMS, MINSK, REPUBLIC OF BELARUS;
[4]Zazhogin Stanislav Dmitrievich – Senior Software Engineer,
International IT Integrator Hospitality & Retail Systems;
[5]Bobrov Andrei Vladimirovich – Team leader,
TECHNICAL SUPPORT GROUP,
SHARXDC LLC,
MOSCOW

Abstract: *methods and models for the safe use of cloud computing in mobile applications are considered. The importance of the problem of organizing authentication mechanisms within a cloud service is shown. The features of the organization of stable operation of a mobile cloud service are determined. A security model has been developed for a mobile cloud service providing database services (DBaaS). The DBaaS service schema includes four levels: user interface level, application level, database level, and data storage level. At the same time, the proposed security model is based on the adaptation and expansion of the algorithms of the Needham-Schroeder's protocol for confirming the authorization of a cloud service user of the DBaaS type. The analysis showed the effectiveness of the developed model in relation to the standard cyber threats of cloud services.*
Keywords: *cloud service, mobile application, cyber threat, user authentication, database as a service (DBaaS), Needham-Schroeder's protocol.*

РАЗРАБОТКА АУТЕНТИФИКАЦИОННОЙ МОДЕЛИ ДЛЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Вишняков А.С.[1], Макаров А.Е.[2], Уткин А.В.[3], Зажогин С.Д.[4], Бобров А.В.[5] (Российская Федерация)

[1]Вишняков Александр Сергеевич – ведущий инженер,
системный интегратор «Крастком»;
[2]Макаров Анатолий Евгеньевич – архитектор решений,
Российская телекоммуникационная компания «Ростелеком»,
г. Москва;
[3]Уткин Александр Владимирович – старший инженер,
Международный системный интегратор «EPAM Systems», г. Минск, Республика Беларусь;
[4]Зажогин Станислав Дмитриевич - старший разработчик,
Международный IT интегратор «Hospitality & Retail Systems»;
[5]Бобров Андрей Владимирович – руководитель группы,
группа технической поддержки,
Компания SharxDC LLC,
г. Москва

Аннотация: *рассмотрены методы и модели безопасного использования облачных вычислений в мобильных приложениях. Показана значимость проблемы организации механизмов аутентификации в рамках облачного сервиса. Определены особенности организации стабильной работы мобильного облачного сервиса. Разработана модель обеспечения безопасности мобильного облачного сервиса, предоставляющего услуги базы данных (DBaaS). Схема DBaaS-службы включает в себя четыре уровня: уровень пользовательского интерфейса, уровень приложения, уровень базы данных и уровень хранилища данных. При этом предложенная модель обеспечения безопасности базируется на адаптации и расширении алгоритмов протокола Нидхэма-Шредера подтверждения авторизации пользователя облачного сервиса типа DBaaS. Проведенный анализ показал эффективность разработанной модели по отношению к стандартным кибер-угрозам облачных сервисов.*
Ключевые слова: *облачный сервис, мобильное приложение, кибер-угроза, аутентификация пользователя, база данных как сервис (DBaaS), протокол Нидхэма-Шредера.*

УДК 331.225.3

Введение: Внедрение парадигмы облачных вычислений в мобильных приложениях позволило разработчикам исключить привязку к мобильной операционной системе (ОС) и аппаратной платформе мобильного устройства. Доступ к мобильным облачным вычислениям осуществляется через мобильный браузер удаленного веб-сервера, причем в ряде случае это может быть реализовано без установки пользователем клиентского приложения.

Современные мобильные сети обеспечивают эффективную инфраструктуру с возможностью доступа к большим базам данных (БД). Таким образом, актуализируется облачный сервис типа DBaaS (БД как сервис), равно как и защита таких сервисов от утечки конфиденциальной информации. В результате этого вырастает значимость механизмов аутентификация базы данных, процесса подтверждения того, что пользователь, запрашивающий доступ к определенным действиям в отношении БД, имеет на этот доступ право.

DBaaS предоставляет платформу профессиональной базы данных, работа которой может быть организована без предварительного обучения персонала. При этом поставщик услуг в соответствии с опытом обслуживания большого количества клиентов может предложить оптимальную конфигурацию базы данных с минимумом уязвимостей. Сервис упрощает задачи решения типичных проблем работы с большими объемами данных, исправления ошибок, передачу данных между системами и масштабирования платформы.

DBaaS сервис имеет существенные преимущества перед многочисленными аналогами: сервис-ориентированный подход (средства БД предоставляются как сервис) и организация клиентом собственной модели работы с БД без приобретения дополнительного оборудования и программных продуктов, что обуславливает ***актуальность разработки алгоритмов аутентификации*** в рамках облачного сервиса данного типа.

Анализ последних исследований и публикаций в данной области показал перспективность применения облачных вычислений в мобильных приложениях. Помимо DBaaS были рассмотрены типичные угрозы других типов облачных сервисов, а также методы организации их безопасности и конфиденциальности [1, 2]. Были изучены методики безопасной передачи данных, в частности, методики, предложенные в работах [3, 4], которые включают в себя безопасный механизм передачи данных (SDTM: Secure Data Transmission Mechanism), а также систему обнаружения вредоносных пакетов (MPDS: Malicious Packets Detection System).

В соответствии с исследованиями [5, 6] была обоснована небезопасность использования промежуточных компонент доступа к базе данных со стороны клиента (прокси-сервер клиента) как потенциальной единой точки отказа DBaaS-службы. Также был проведен анализ использования облачными сервисами типа DBaaS сторонних служб для сохранения целостности данных, которые передаются сторонним поставщикам DBaaS [7, 8].

Было показано, что в рамках данного подхода пользователи получают доступ к услугам по требованию, не сталкиваясь с необходимостью обслуживания локального хранилища данных. Дальнейший анализ включал изучение схем безопасного обслуживания данных [9, 10] с моделями повторного шифрования (PRE: Proxy Re-Encryption) и шифрования на основе идентификатора (IDE: Identity Based Encryption), что подразумевает выполнение процедуры криптографии на уровне пользователя.

В результате проведенного анализа показан приоритет в области обеспечения кибер-безовасности мобильных облачных сервисов алгоритмов, применяемых в протоколе Нидхэма-Шредера [11, 12].

Целью работы, таким образом, стала разработка комплексной методики защиты мобильных служб облачных сервисов типа DBaaS при помощи алгоритмов аутентификации алгоритмов, которые базируются на протоколе Нидхэма-Шредера.

1. Особенности организации работы облачной DBaaS-службы

DBaaS — это служба, управляемая оператором облачного сервиса, ориентированная на поддержку пользовательских приложений в части предоставления работы с БД. Таким образом, с разработчиков мобильных приложений снимается необходимость администрирования и обслуживания БД, что является существенным преимуществом сервисов данного типа. БД DBaaS-службы должна поддаваться эффективному масштабированию, автоматически обновляться, создавать резервные копии и обрабатывать аппаратные сбои без ущерба для пользователя. Таким образом, для предоставления полноценной организации DBaaS-службы, которая работает с динамично растущим количеством пользовательских приложений, необходимо решить вопросы автоматизации и кибер-защиты. При этом, в пользовательском соглашении (SLA: Service Level Agreement) указываются требования по уровню обслуживания (QoS: Quality of Service) с учетом количества соединений, пиковым уровнем загрузки центрального процессора (ЦП) и других критериев, а за счет виртуализации ресурсов программно-аппаратной платформы пользователь способен с минимальными затратами времени внести в SLA необходимые изменения [5, 6]. Внедрение DBaaS может существенно снизить эксплуатационные расходы и при сохранении прежнего уровня эффективности работы приложений за счет обеспечения:

- консолидированной платформы общего доступа;
- гибкой модели самоорганизации и самообслуживания структуры БД;
- эластичности в вопросах масштабирования и уменьшения ресурсоемкости БД;
- соотнесения затрат с реальным уровнем эксплуатации БД.

При этом следует отметить, что системы управления БД (СУБД) в целом не являются компонентом, который поддается эффективному масштабированию стандартными программными средствами мобильных приложений. Так, например, при анализе облачного сервиса «Amazon Relational Database Service» можно выделить три основных проблемы проектировки распределенных реляционных баз данных [13]:

1. организация эффективной многопользовательской работы в условиях прогнозируемой рабочей нагрузки на серверный комплекс;
2. масштабирование комплекса в условиях роста рабочей нагрузки;
3. обеспечение конфиденциальности информации, хранящейся в БД.

В то же время в ряде работ отмечает, что эффективность дальнейшего внедрения парадигмы облачных вычислений критически зависит от создания масштабируемых, автономны и безопасных СУБД, которые обеспечивают функционал высокого уровня и согласованность работы.

Таблица 1. Особенности организации стабильной работы облачного сервиса типа DBaaS

Требования к DBaaS-платформе	**Потенциальные угрозы**
Обеспечение бесперебойной работы программно-аппаратного комплекса	Проблемы программного обеспечения
	Выход из строя оборудования серверов и сетевого оборудования
	Превышение нагрузки на сервера
	Форс-мажорные обстоятельства
Обеспечение пользователями и персоналу надлежащих прав доступа к сервису	Недостаточный контроль прав доступа
	Высокозатратные и недостаточно гибкие средства обеспечения контроля
Обеспечение информационной целостности сервиса	Изменение конфигурации системы, нарушение согласованности управления
	Нарушение целостности данных
	Отсутствие истории изменения данных и настроек сервисов
Обеспечение проверок и мониторинга	Отсутствие резервного копирования
	Отсутствие алгоритмов переключения в аварийном режиме работы и восстановления поврежденных данных
	Отсутствие эффективного мониторинга
Обеспечение конфиденциальности данных сервиса	Отсутствие алгоритмов шифрования
	Совместные информационные ресурсы
Защита сервиса от потенциальных внутренних угроз	Некомпетентность и диверсии разработчиков ПО
	Некомпетентность и диверсии персонала
Защита сервиса от потенциальных внешних угроз	Кибер-атака на ресурсы аппаратно-программного комплекса: вирусы, скрытые каналы, троянские программы, DoS/DDoS-атаки
	Проблемы, связанные с некорректными действиями пользователей

В табл. 1 приведены результаты анализа особенностей организации стабильной работы облачного сервиса типа DBaaS с выявлением требований к соответствующей платформе и потенциальных угроз. Следует отметить, что потенциальные угрозы DBaaS-службы не являются взаимозависимыми, поэтому связи между отдельными компонентами показаны отдельно на рис. 1.

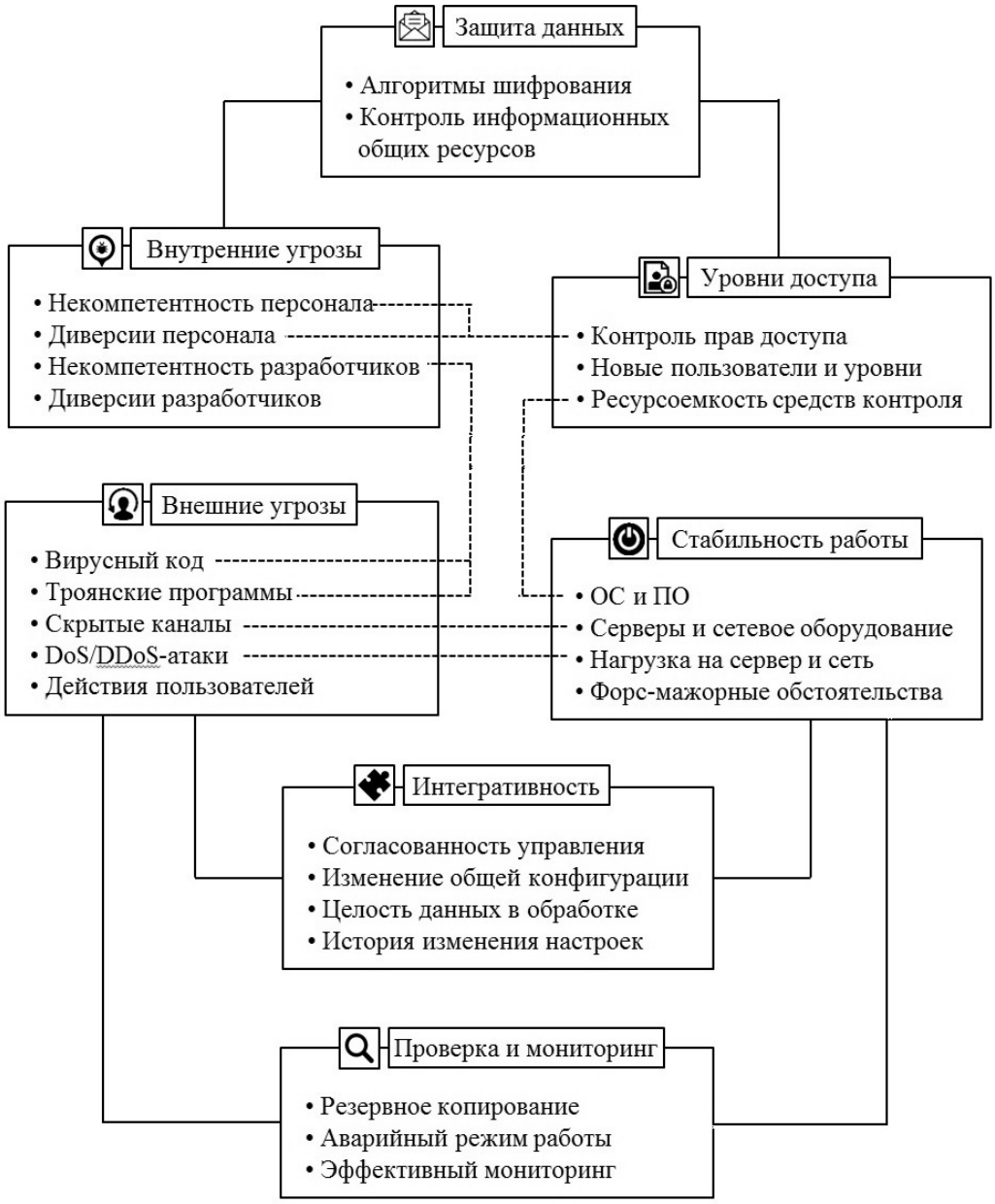


Рис. 1. Соотнесение требований к DBaaS-службе и потенциальных угроз

Представленная схема соотнесения требований к DBaaS-службе и потенциальных угроз позволяет построить модель работы соответствующего мобильного облачного сервиса и построить комплексный алгоритм обеспечения стабильности его функционирования.

2. Модель обеспечения аутентификации пользователей DBaaS-службы

Базовая схема функционирования DBaaS-службы включает в себя четыре уровня: уровень пользовательского интерфейса, уровень приложения, уровень базы данных и уровень хранилища данных (рис. 2), функциональные компоненты которых могут быть определены на основе результатов анализа, проведенного в предыдущем разделе.

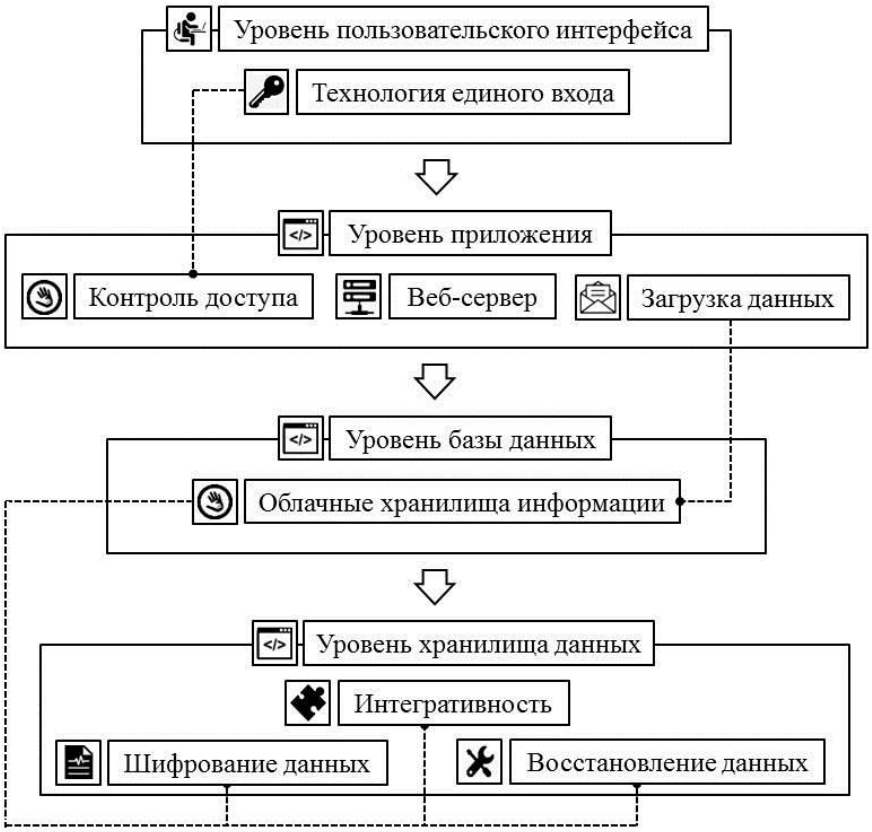


Рис. 2. Четырехуровневая схема функционирования DBaaS-службы

В рамках данного исследования в основу модели аутентификации пользователей мобильного облачного сервиса было предложено положить протокол Нидхэма-Шредера [11, 12]. Данный протокол использует открытый ключ для достижения аутентификации между двумя участниками с помощью центра аутентификации. В основу построения модели протокола включаются такие условные обозначения как:

- U — пользователь;
- S — сервер;
- C — удостоверяющий центр (CA: Certification Authority), который подт-верждает подлинность ключей шифрования.

Данные элементы берутся в основу следующих функций, определяющих использование открытых и секретных ключей аутентификации, а также одноразового кода, который выбирается случайным или псевдослучайным образом:

- $K_p(C)$ — открытый ключ (public key) удостоверяющего центра;
- $K_S(C)$ — секретный ключ (secret key) удостоверяющего центра;
- $K_p(U)$ — открытый ключ пользователя;
- $K_p(S)$ — открытый ключ сервера;
- $K_{SS}(U, C)$ — совместный ключ сессии (shared session key) пользователя и удостоверяющего центра;
- $K_{SS}(S, C)$ — совместный ключ сессии сервера и удостоверяющего центра;
- $N(U)$ — одноразовый код (nonce), выбранный случайным (псевдо-случайным) образом пользователем;
- $N(S)$ — одноразовый код, выбранный случайным (псевдослучайным) образом сервером

На основе данных компонентов была выстроена модель аутентификации мобильного облачного сервиса, которая включает семь этапов, представленная в табл. 2 с соответствующими обозначениями,

которые в дальнейшем будут использованы в диаграммах. Многоэтапная методика проверки ключей двух сторон (пользователь и сервис) через удостоверяющий центр позволяет обеспечить надежную защиту мобильного облачного сервиса от внешних кибер-угроз.

Тем не менее, на сегодняшний день существует методика обхода данной защиты. Модель такой кибер-атаки может быть построена через введение дополнительного компонента, соответствующего злоумышленнику, который выдает себя за сервер (intruder), а также функций, которые соответствуют ключам, которые использует данный злоумышленник с целью обхода защитных алгоритмов протокола Нидхэма-Шредера:

- I — злоумышленник, который выдает себя за сервер;
- $K_{SS}(I, C)$ — совместный ключ сессии злоумышленника и удостоверяющего центра;
- $K_{SS}(I, U)$ — совместный ключ сессии злоумышленника и пользователя;
- $K_P(I)$ — открытый ключ злоумышленника.

Таблица 2. Базовая модель аутентификации пользователей мобильного облачного сервиса на основе протокола Нидхэма-Шредера

№	Этап	Обозначение
1	Пользователь запрашивает открытый ключ у удостоверяющего центра.	$U \to C: K_P(S)$
2	Удостоверяющий центр высылает заверенный цифровой подписью открытый ключ и идентификатор пользователю.	$C \to U: \{K_P(S), K_{SS}(S, C)\} \cdot K_S(C)$
3	Пользователь высылает серверу одноразовый код и идентификатор.	$U \to S: \{K_P(U), N(U)\} \cdot K_{SS}(C, U)$
4	Сервер запрашивает открытый ключ пользователя у удостоверяющего центра.	$S \to C: K_P(U)$
5	Удостоверяющий центр высылает серверу открытый ключ и идентификатор.	$C \to S: \{K_P(U), K_{SS}(U, C)\} \cdot K_P(C)$
6	Сервер генерирует одноразовый код и передает их вместе с кодом пользователя, зашифровав через открытый ключ пользователя.	$S \to U: N\big(U, K_P(U)\big),$ $N(S, K_P(U))$
7	Пользователь высылает обратно серверу одноразовый код, который при этом зашифрован открытым ключом сервера.	$U \to S: N(S, K_{SS}(S, C))$

Модель обхода алгоритмов аутентификации представлена в табл. 3. Как можно видеть, данная модель соответствует модели, представленной в табл. 2, но в нее добавлены пункты, которые соответствуют деятельности злоумышленника, поочередно выдающего себя за сервер и пользователя (данные этапы в таблице выделены серым).

Таблица 3. Модель обхода алгоритмов аутентификации мобильного облачного сервиса на основе протокола Нидхэма-Шредера

№	Этап	Обозначение
1	Пользователь запрашивает открытый ключ у удостоверяющего центра.	$U \to C: K_P(S)$
2	Удостоверяющий центр высылает заверенный цифровой подписью открытый ключ и идентификатор пользователю.	$C \to U: \{K_P(S), K_{SS}(S, C)\} \cdot K_S(C)$

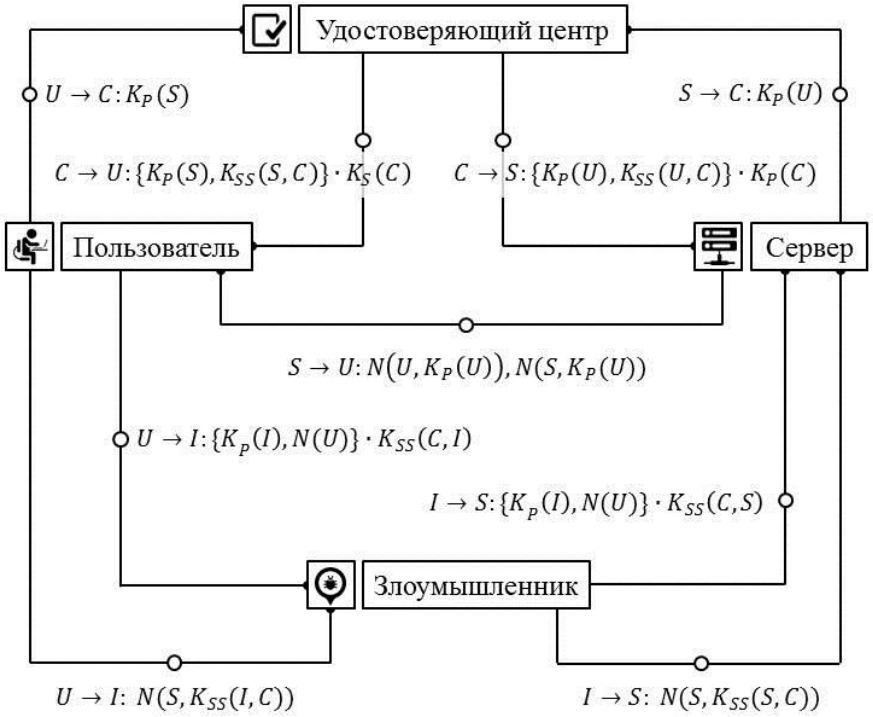
3	Пользователь высылает злоумышленнику одноразовый код и идентификатор.	$U \to I: \{K_P(I), N(U)\} \cdot K_{SS}(C, I)$
4	Злоумышленник высылает серверу одноразовый код и идентификатор.	$I \to S: \{K_P(I), N(U)\} \cdot K_{SS}(C, S)$
5	Сервер запрашивает открытый ключ пользователя у удостоверяющего центра.	$S \to C: K_P(U)$
6	Удостоверяющий центр высылает серверу открытый ключ и идентификатор.	$C \to S: \{K_P(U), K_{SS}(U, C)\} \cdot K_P(C)$
7	Сервер генерирует одноразовый код и передает их вместе с кодом пользователя, зашифровав через открытый ключ пользователя.	$S \to U: N\big(U, K_P(U)\big),$ $N(S, K_P(U))$
8	Пользователь высылает злоумышленнику одноразовый код серверах зашифрованный совместным ключом сессии злоумышленника и удостоверяющего центра.	$U \to I: N(S, K_{SS}(I, C))$
9	Злоумышленник высылает серверу одноразовый код серверах зашифрованный совместным ключом сессии сервера и удостоверяющего центра..	$I \to S: N(S, K_{SS}(S, C))$

Таким образом, адаптация рассмотренных алгоритмов аутентификации должна включать общий ключ между пользователем и удостоверяющим центром $K(U, C)$, а также двух последовательностей одноразового кода, выбранного случайным (псевдослучайным) образом сервером $(N_1(U), N_2(U)$ и пользователем ($N_1(S), N_2(S)$), соответственно (табл. 4).

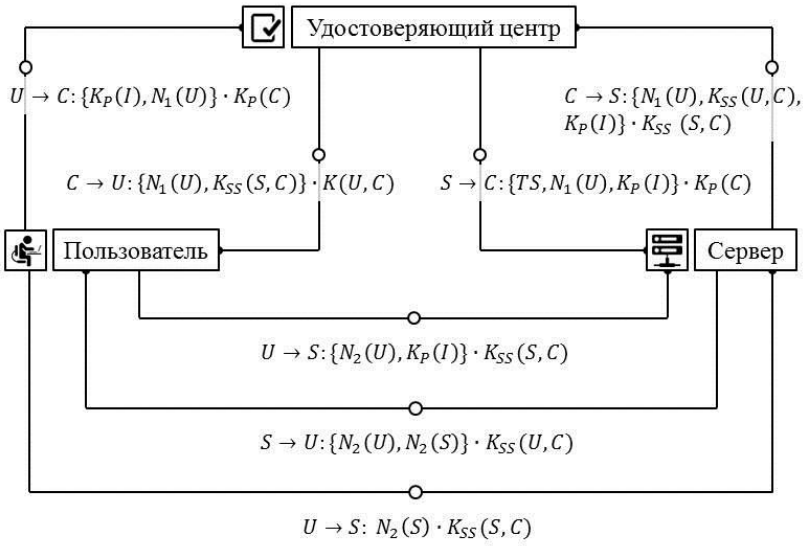
Таблица 4. Предложенная модель аутентификации пользователей мобильного облачного сервиса на основе протокола Нидхэма-Шредера

№	Этап	Обозначение
1	Пользователь, используя общий ключ, а также последовательность одноразового кода, выбранного случайным или псевдослучайным образом, получает у удостоверяю-щего центра, заверенный цифровой подписью открытый ключ и идентификатор	$U \to C: \{K_P(I), N_1(U)\} \cdot K_P(C)$
2		$C \to U: \{N_1(U), K_{SS}(S, C)\} \cdot K(U, C)$
3	Пользователь высылает серверу второй одноразовый код и идентификатор.	$U \to S: \{N_2(U), K_P(I)\} \cdot K_{SS}(S, C)$

4	Сервер запрашивает открытый ключ пользователя у удостоверяющего центра.	$S \rightarrow C: \{TS, N_1(U), K_P(I)\} \cdot K_P(C)$
5	Удостоверяющий центр высылает серверу открытый ключ и идентификатор.	$C \rightarrow S: \{N_1(U), K_{SS}(U,C), K_P(I)\} \cdot K_{SS}(S,C)$
6	Сервер генерирует одноразовый код и передает их вместе со вторым кодом пользователя, зашифровав через открытый ключ пользователя.	$S \rightarrow U: \{N_2(U), N_2(S)\} \cdot K_{SS}(U,C)$
7	Пользователь высылает обратно серверу второй одноразовый код сервера, зашифрованный открытым ключом сервера.	$U \rightarrow S: N_2(S) \cdot K_{SS}(S,C)$



(a)



(б)

Рис. 3. Соотнесение методики обхода алгоритмов аутентификации (а) и усовершенствованной модели аутентификации пользователей сервиса (б).

На рис. 3 представлено соотнесение методики обхода алгоритмов аутентификации пользователей мобильного облачного сервиса, разработанной на основе протокола Нидхэма-Шредера (рис. 3-а) и усовершенствованной модели, которая блокирует действия потенциального злоумышленника (рис. 3-б). Уже на этапах 1 и 2 (табл. 4) алгоритмы аутентификации не дают злоумышленнику возможность ввести пользователя в заблуждение, поскольку на этих этапах происходит шифрование при помощи общего ключа с удостоверяющим центром. На третьем этапе злоумышленник также не может отправить одноразовый код и выдать себя пользователем, подающим запрос серверу. Соответственно, если злоумышленник получает передаваемые данные на этапах 5-7, у него не остается возможности расшифровать содержимое.

Выводы

Рассмотрены методы внедрения парадигмы облачных вычислений в мобильных приложениях. Показана необходимость построения эффективных алгоритмов аутентификации пользователей с целью увеличения уровня безопасности соответствующей службы. Разработана четырехуровневая схема организации DBaaS-службы, которая включает в себя уровни пользовательского интерфейса, приложения, базы данных и хранилища данных. Предложена модель обеспечения безопасности сервиса, которая базируется на использовании алгоритмов протокола Нидхэма-Шредера. Проведенный анализ модели показал эффективность работы данной модели по сравнению со стандартным протоколом Нидхэма-Шредера.

Список литературы / References

1. *Asija R., Nallusamy R.* Healthcare SaaS based on a data model with built-in security and privacy. Int. J. Cloud Appl. Comput. 6 (3), 2016.
2. *Almorsy M., Ibrahim A. & Grundy J.*, 2013. Adaptive Security Management in SaaS Applications. Security, Privacy and Trust in Cloud Systems, 73-102. doi:10.1007/978-3-642-38586-5_3.
3. *Alhaj A., Aljawarneh S., Masadeh S., Abu-Taieh E.* A secure data transmission mechanism for cloud outsourced data. Int. J. Cloud Appl. Comput. 3 (1), 34–43, 2013.
4. Alhaj A.A., 2015. Performance Evaluation of Secure Data Transmission Mechanism (SDTM) for Cloud Outsourced Data and Transmission Layer Security (TLS). Cloud Technology, 839-844. doi:10.4018/978-1-4666-6539-2.ch038.
5. *Ferretti L., Colajanni M., Marchetti M*. Supporting security and consistency for cloud database, cyberspace safety and security. *Lecture Notes in Computer Science*, Vol. 7672, Pp. 179–193 (2012).
6. *Munir K.*, 2019. Security Model for Mobile Cloud Database as a Service (DBaaS). Cloud Security, 760-769. doi:10.4018/978-1-5225-8176-5.ch038.
7. *Cong W., Sherman S.M.C., Qian W., Kui R., Wenjing* L. Privacy preserving public auditing for secure cloud storage. IEEE Trans. Comput. 62(2), 362–375, 2013.
8. Public Auditing for a Secure Cloud Storage using Dynamic Hash Table, 2017. International Journal of Recent Trends in Engineering and Research, 3(12), 14-20. doi:10.23883/ijrter.2017.3529.mmycv.

9. *Jia W., Zhu H., Cao Z., Wei L., Lin X*. SDSM: a secure data service mechanism in mobile cloud computing. In: Proceedings of the IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, Shanghai, China, 2011.
10. *Yajam H.A., Mohajeri J. & Salmasizadeh M*., 2015. Identity-based universal re-encryption for mixnets. Security and Communication Networks, 8(17), 2992-3001. doi:10.1002/sec.1226.
11. *Xiao M., Deng C., Ma C., Zhu K. & Cheng D.*, 2015. Proving Authentication Property of Modified Needham-Schroeder Protocol with Logic of Events. Proceedings of the International Conference on Computer Information Systems and Industrial Applications. doi:10.2991/cisia-15.2015.103.
12. *Liu S.M., Ye J.Y. & Wang Y.L*., 2014. Improvement and Security Analysis on Symmetric Key Authentication Protocol Needham-Schroeder. Applied Mechanics and Materials, 513-517, 1289-1293. doi:10.4028/www.scientific.net/amm.513-517.1289.
13. *Wittig M., Wittig A. & Whaley B.*, 2019. Amazon Web Services in action. Shelter Island, NY: Manning.