ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
INFORMATION TECHNOLOGIES

УДК 004.056.52
DOI: 10.37661/1816-0301-2025-22-2-95-110

Оригинальная статья
Original Article

Прототипирование системы беспарольного атрибутивного доступа к информационным ресурсам с использованием eID-карты Республики Беларусь и FIDO2-аутентификации

А. А. Жидович, А. А. Лубенько, И. С. Войтешенко✉

Белорусский государственный университет,
пр. Независимости, 4, Минск, 220030, Беларусь
✉*E-mail: voit@bsu.by*

Аннотация

Цели. Целью проведенной аналитической и исследовательской работы являются проектирование и реализация прототипа системы установки личности пользователя и его привилегий путем совместного использования беспарольной FIDO2-аутентификации и управления доступом на основе атрибутов. В качестве источника пользовательских атрибутов предложены средства электронной идентификации, соответствующие стандартам ICAO.

Методы. В исследовании применялись: систематизация и анализ литературы и технических спецификаций; системный подход к анализу существующих реализаций систем беспарольного атрибутивного доступа и теоретических моделей, используемых при их проектировании; SCn- и SCg-код технологии OSTIS для семантического описания основных понятий и концепций, связанных с FIDO2-аутентификацией; программные платформы и библиотеки.

Результаты. Результатом работы является прототип системы атрибутивного доступа к информационным ресурсам в цифровой среде с использованием eID-карты Республики Беларусь и FIDO2-аутентификации. Разработанное приложение было контейнеризовано и развернуто на онлайн-сервере, его работоспособность проверена с различных платформ с помощью распространенных браузеров.

Заключение. Представлено исследование по разработке и первоначальной оценке прототипа системы управления доступом к информационным ресурсам с помощью аутентификации по спецификации FIDO2 и модели управления доступом на основе атрибутов. При этом в качестве источника пользовательских атрибутов применяются средства электронной идентификации, удовлетворяющие стандартам Международной организации гражданской авиации, в том числе eID-карта Республики Беларусь.

Ключевые слова: беспарольная аутентификация, авторизация на основе атрибутов, спецификация FIDO2, протокол W3C WebAuthn, протокол «клиент-аутентификатор», eID-идентификация

Благодарности. Авторы выражают благодарность за содействие в работе профессору А. Н. Курбацкому и сотрудникам ЗАО «АВЕСТ».

Для цитирования. Жидович, А. А. Прототипирование системы беспарольного атрибутивного доступа к информационным ресурсам с использованием eID-карты Республики Беларусь и FIDO2-аутентификации / А. А. Жидович, А. А. Лубенько, И. С. Войтешенко // Информатика. – 2025. − Т. 22, № 2. – С. 95–110. – DOI: 10.37661/1816-0301-2025-22-2-95-110.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 17.03.2025
Подписана в печать | Accepted 01.04.2025
Опубликована | Published 30.06.2025

Prototyping of the system of passwordless attributive access to information resources using eID-card of the Republic of Belarus and FIDO2-authentication

Anton A. Zhidovich, Alexei A. Lubenko, Iosif S. Vojteshenko✉

Belarusian State University,
av. Nezavisimosti, 4, Minsk, 220030, Belarus
✉*E-mail: voit@bsu.by*

Abstract

Objectives. The purpose of the analytical and research work carried out is to design and implement a prototype system for establishing user identity and privileges based on the joint use of passwordless FIDO2 authentication and attribute-based access control. It is proposed that electronic identification means compliant with ICAO standards be used as a source of user attributes.

Methods. The following were used in this paper: systematization and analysis of literature and technical specifications; systematic approach to the analysis of existing implementations of passwordless attribute access systems and theoretical models used in their design; the SCn- and SCg-code of OSTIS technology for semantic description of basic concepts and concepts related to FIDO2-authentication; software platforms and libraries.

Results. The result of the work is a prototype of the system of attributive access to information resources in the digital environment using the eID-card of the Republic of Belarus and FIDO2-authentication. The developed application was containerized and deployed on the online server. Its performance was then tested from different platforms using standard browsers.

Conclusion. A study on the development and initial evaluation of a prototype of an information resource access control system based on authentication to the FIDO2 specification and an attribute-based access control model is presented. At the same time, as a source of user attributes the means of electronic identification that meet the standards of the International Civil Aviation Organization, including the eID-card of the Republic of Belarus, are used.

Keywords: passwordless authentication, attribute-based authorization, FIDO2 specification, protocol W3C WebAuthn, client-to-authenticator protocol, eID-identification

Acknowledgments. The authors express their gratitude to professor A. N. Kurbatsky and the staff of CJSC AVEST for their assistance in this work.

For citation. Zhidovich A. A., Lubenko A. A., Vojteshenko I. S. *Prototyping of the system of passwordless attributive access to information resources using eID-card of the Republic of Belarus and FIDO2-authentication.* Informatika [*Informatics*], 2025, vol. 22, no. 2, pp. 95−110 (In Russ.). DOI: 10.37661/1816-0301-2025-22-2-95-110.

Conflict of interest. The authors declare of no conflict of interest.

Введение. Для большинства киберпреступлений характерна кража учетных данных пользователя. Например, получив доступ к учетным данным жертвы, злоумышленник может продать данные или совершить противоправные действия различной направленности.

В основе систем безопасности в цифровой среде лежит процесс установки личности пользователя и его привилегий. Таким образом, вопросы идентификации, аутентификации и авторизации являются одними из самых важных при построении безопасной информационной системы.

Современные многокомпонентные системы, состоящие из множества взаимодействующих ресурсов, характеризуются высокой сложностью в контексте требований к авторизации пользователей. Организация эффективного и безопасного механизма установки личности пользователя и его привилегий зачастую становится приоритетной задачей для разработчиков. Сложность системы авторизации обусловливает необходимость создания интеграционных решений, обеспечивающих унификацию подходов к идентификации и управлению доступом. Это требует дополнительных усилий на этапе проектирования, что увеличивает общую стоимость и временны́е затраты на разработку таких систем.

Активное развитие электронных идентификационных документов (eID) открывает новые горизонты для безопасного и эффективного доступа, например, к финансовым, медицинским, муниципальным и государственным информационным системам. В условиях растущей необходимости в защите персональных данных и соблюдении соответствующих требований законодательства интеграция eID в такие системы может стать ключевым шагом к повышению уровня безопасности, удобства и доверия пользователей.

Одним из трендов в информационной безопасности является процесс перехода к беспарольным методам аутентификации. В частности, все большее внимание уделяется спецификациям FIDO2, повышающим безопасность и конфиденциальность пользователя по сравнению с классическими парольными методами. На данный момент в мире ряд организаций предоставляют унифицированный компонент, позволяющий интегрировать FIDO2-аутентификацию в различные платформы. Стоит отметить, что этот подход пользуется доверием среди компаний в области финансовых технологий, где уязвимость может привести к серьезным потерям.

Таким образом, унифицированная система установки личности пользователя и его привилегий, интегрирующая возможности eID-идентификации, гибкой авторизации и FIDO2-аутентификации, способна сократить время и стоимость разработки сложных информационных систем, повысить степень защиты конфиденциальных данных, а также упростить доступ для пользователей.

Целями настоящей работы являются проектирование и реализация прототипа системы установки личности пользователя и его привилегий на основе совместного применения беспарольной FIDO2-аутентификации и атрибутивного управления доступом. В качестве источника пользовательских атрибутов предложено использование средств электронной идентификации, соответствующих стандартам Международной организации гражданской авиации (ICAO). Было показано, что на данный момент это решение является перспективным и удобным в использовании.

Теоретические и технологические предпосылки проведения исследований и их практической реализации

Технология беспарольной FIDO2-аутентификации. В отличие от доступа с помощью пароля беспарольные решения с самого начала разрабатывались с учетом современного удобства использования и современных ландшафтов атак. Примерами беспарольного доступа являются доступы с применением Windows Hello[1], отпечатка пальца, сканера сетчатки глаза, FIDO-токенов и т. д. Все эти решения по своей сути являются многофакторными и безопасными с точки зрения протоколов.

Несмотря на то что технология FIDO2 уже находит достаточно широкое практическое применение, научные исследования и практические разработки в этой области активно развиваются [1–6].

[1]Windows Hello. – URL: https://www.microsoft.com/en-us/windows/tips/windows-hello (date of access: 12.01.2025).

Стандарт FIDO2, разработанный FIDO-альянсом[2], представляет собой совокупность спецификаций в двух частях. Первая часть называется стандартом веб-аутентификации W3C WebAuthn[3]. Эта спецификация представляет собой JavaScript API, с помощью которого удаленные веб-сайты могут запрашивать учетные данные открытого ключа. Вторая часть – это спецификация протокола «клиент-аутентификатор» (client-to-authenticator protocol, CTAP). Протокол «клиент-аутентификатор» описывает, как приложение и операционная система устанавливают связь с совместимым устройством аутентификации через USB-, NFC- или BLE-средства связи.

Взаимодействие между платформой и аутентификатором по данному протоколу можно описать следующим образом:

1. Платформа устанавливает соединение с аутентификатором.
2. Платформа получает информацию об аутентификаторе и его возможностях.
3. Платформа отправляет команду для операции, поддерживаемой средством аутентификации.
4. Аутентификатор отвечает данными ответа или ошибкой.

Технология, лежащая в основе FIDO2, базируется на асимметричной криптографии. В этом процессе каждый закрытый и открытый ключ генерирует пару ключей. Открытый ключ получателя не является секретом и может быть обменен или опубликован в любое время с партнерами по коммуникации для шифрования сообщений. Для расшифровки сообщения требуется закрытый ключ. Он должен храниться получателем в безопасности, поскольку третьи стороны могли бы расшифровать сообщение, если бы они его перехватили.

Каждый аутентификатор, доступный для FIDO2, имеет сертификат X.509, также известный как сертификат аттестации, сохраняемый при изготовлении устройства. Закрытый ключ записывается на устройство и не может быть экспортирован или иным образом изменен. Кроме того, устройство криптографически сертифицировано. Это означает, что, если злоумышленники попытаются перехватить запрос на регистрацию и заменить его своим собственным, они не смогут обменять сгенерированный открытый ключ на другой ключ, поскольку подписи подтверждения не будут совпадать. Схема взаимодействия компонентов FIDO2 изображена на рис. 1.

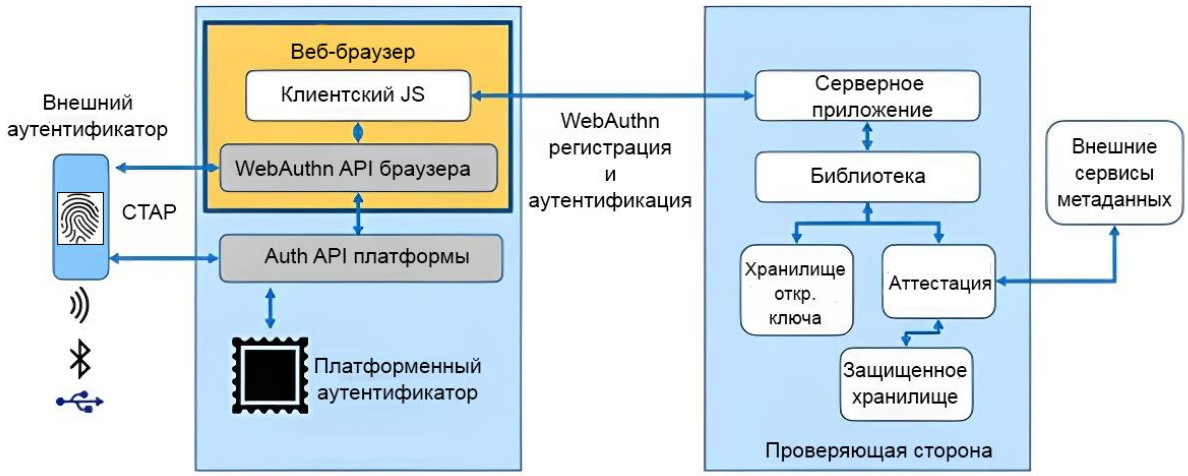


Рис. 1. Компоненты FIDO2 и их взаимодействие

Fig. 1. FIDO2 components and their interaction

[2]Fido Alliance. – URL: https://fidoalliance.org (date of access: 12.01.2025).
[3]WebAuthn API specification. – URL: https://w3c.github.io/webauthn/ (date of access: 12.01.2025).

Платформозависимый и платформонезависимый аутентификаторы действуют как защищенный объект, интегрированный в конечное устройство пользователя (например, Windows Hello в качестве платформозависимого аутентификатора) или подключенный к нему (например, аппаратный токен USB в качестве платформонезависимого аутентификатора). Служба метаданных FIDO помогает подтвердить подлинность аутентификатора, регулярно извлекая цепочки сертификатов аутентификатора.

Производители генерируют и хранят глобально уникальный идентификатор подтверждения подлинности случайным образом, который идентифицирует модель аутентификации. Процесс регистрации уникального идентификатора (ключа) подробно описан в работе [4].

Для аутентификации пользователь подтверждает свою личность проверяющей стороне через клиента WebAuthn с помощью средства аутентификации.

После того как пользователь зарегистрировал свой ключ, он может применять его для беспарольной аутентификации. Порядок действий при аутентификации во многом совпадает с процессом регистрации, однако имеет свои особенности [7]. Рассмотрим подробнее процесс аутентификации:

1. Когда пользователь переходит к интернет-сервису, поддерживающему WebAuthn, и хочет пройти аутентификацию, для клиента WebAuthn отправляется запрос на аутентификацию проверяющей стороне.
2. Проверяющая сторона получает запрос на аутентификацию и отправляет сгенерированный запрос обратно клиенту WebAuthn.
3. Клиент перенаправляет полученный запрос аутентификатору. В запросе содержатся информация о необходимом типе аутентификатора, идентификаторы зарегистрированных ключей и идентификатор пользователя.
4. Аутентификатор находит сохраненные ключи и предлагает пользователю выбрать один из них. После этого выполняется проверка подлинности пользователя. В случае успеха аутентификатор создает ответ, называемый assertion response, который содержит в себе подпись, созданную закрытым ключом, и другую информацию, необходимую для проверяющей стороны.
5. Аутентификатор отправляет ответ клиенту.
6. Клиент создает и отправляет ответ, содержащий в себе assertion response, проверяющей стороне.
7. Проверяющая сторона проверяет доступность открытого ключа в базе данных и подпись, находящуюся в assertion response, с помощью открытого ключа, сохраненного во время регистрации.
8. Проверяющая сторона дает доступ к данным пользователя.

Совместное использование FIDO2-аутентификации и управления доступом на основе атрибутов может существенно повысить уровень гибкости и масштабируемости системы безопасности. Важно понимать, что это два разных процесса, причем базовая спецификация FIDO2 (в частности, WebAuthn) не предусматривает подобную интеграцию. Следовательно, совместная настройка ABAC и FIDO2 может требовать дополнительных усилий и ресурсов, специфических для каждой системы или задачи. Например, если в системе присутствуют динамические атрибуты (такие, как местоположение), то интеграция с FIDO2-аутентификацией потребует внедрения механизмов сбора и обновления этих данных, при этом требования спецификации не нарушаются.

Беспарольный доступ на основе атрибутов с использованием медиатора. В работе [2] предложено комплексное и готовое к интеграции в промышленные системы решение для использования анонимных учетных данных с локальной или удаленной аттестацией в виде фреймворка FIDO-AC, который является расширением базовой спецификации FIDO2. Там же представлены оценка безопасности и конфиденциальности, предоставляемых полученной системой, и реализация рабочего прототипа.

Суть полученного в работе [2] решения заключается в создании дополнительной стороны – посредника (медиатора), который отвечает за валидацию данных о пользователе, полученных, например, с eID-документа. Таким образом, в FIDO-AC выделены следующие стороны – участники процесса авторизации:

– FIDO-сервер (проверяющая сторона), формирующий запросы к FIDO-клиенту согласно спецификации FIDO2 и проводящий верификацию полученного ответа;

– FIDO-аутентификатор, использующийся для работы базовой спецификации FIDO2;

– клиент (браузер/приложение с поддержкой WebAuthn) для взаимодействия FIDO-сервера с аутентификатором и приложением FIDO-AC;

– медиатор (посредник), отвечающий за аттестацию данных, полученных с eID;

– приложение FIDO-AC для считывания данных с электронного идентификатора пользователя;

– электронный идентификатор (eID) для удостоверения личности со встроенным запоминающим устройством (чипом), хранящим данные пользователя и подпись эмитента.

Медиатор и приложение FIDO-AC могут быть как отдельными сторонами, так и реализованными модулями одного приложения.

Процесс начинается со считывания пользователем зашифрованных данных и криптографической подписи со своего eID с помощью приложения FIDO-AC. В случае с электронным паспортом требуется ввести номер документа, дату своего рождения и срок действия, на основании которых генерируется ключ для прохождения базового контроля доступа.

Затем при попытке доступа к защищенному ресурсу системы FIDO-сервер создает запрос к клиенту. Он должен соответствовать спецификации WebAuthn и содержать криптографически случайный буфер байтов challenge. Получив запрос, клиент перенаправляет его в приложение FIDO-AC вместе с данными, полученными с электронного документа (а именно данные пользователя, подпись эмитента и открытый ключ, связанный с данными). После этого медиатор должен провести пассивную и активную фазы аутентификации.

На следующем шаге приложение FIDO2 с использованием доказательства с нулевым разглашением (авторами предложены ZK-SNARK и ZK-STARK [8]) на основе данных с eID создает подпись, подтверждающую соответствие пользователя политике ресурса. Эта подпись, а также подпись медиатора передаются клиенту и добавляются к буферу challenge. Затем происходит стандартная FIDO2-аутентификация. Полученный в процессе объект аттестации передается проверяющей стороне.

На завершающем этапе проверяющая сторона должна провести две верификации: проверку подписи, изданной аутентификатором FIDO, и подписи медиатора. Так как значение challenge было изменено, проверяющая сторона должна применить аналогичные изменения к буферу challenge, с которым производится сравнение, а именно присоединить хеш-значение подписи медиатора.

Предлагаемое решение на основе интеграции eID-идентификации, FIDO2-аутентификации и авторизации на основе атрибутов

Функциональные и нефункциональные требования. В соответствии со спецификацией WebAuthn API приложение должно поддерживать регистрацию ключа (платформозависимого) и аутентификацию с его использованием. Прежде всего для регистрации FIDO2-аутентификатора пользователь должен быть авторизованным, т. е. приложение должно иметь возможность аутентификации каким-либо из классических способов. В качестве такого способа был выбран парольный. После регистрации ключа его можно применять для беспарольного доступа в свой аккаунт. При этом желательно, чтобы в результате успешной авторизации устанавливались аутентификационные Cookie. Они создаются на стороне сервера и используются при последующих входах с того же клиента, не требуя от пользователя никаких действий.

Платформа, на которой разработано веб-приложение, должна поддерживать библиотеки, позволяющие выступать ей в роли проверяющей стороны. Они необходимы для создания соответствующих стандарту запросов регистрации ключа и создания подписи для аутентификации с WebAuthn API.

Для того чтобы обеспечить доступ к разработанному приложению с различных устройств, должна быть возможность разместить его на онлайн-сервисе. Также для более удобной разработки желательно, чтобы выбранный сервис поддерживал непрерывное развертывание.

Для развертывания приложение должно поддерживать возможность запуска на UNIX-подобных операционных системах Linux для последующей контейнеризации.

При попытке доступа к защищенному ресурсу, которому соответствует некоторая политика, должно осуществляться считывание данных о пользователе с электронного документа. Для считывания данных электронного документа, проведения валидации представленных на нем данных и проверки соответствия атрибутов пользователя политике, соответствующей конечной точке, разработано клиентское Android-приложение. Оно описано ниже. Таким образом, разрабатываемое веб-приложение должно иметь возможность обращаться к Android-приложению, передавая требуемую политику, а затем получать результат проведенных проверок.

Можно выделить следующие функциональные требования к веб-приложению:

1. Приложение должно иметь хранилище пользовательских данных.
2. Должна быть реализована поддержка работы с политиками доступа к конечным точкам веб-приложения.
3. Должна быть реализована возможность аутентификации пользователя в приложении по паролю с установкой аутентификационных Cookie.
4. В приложении должен быть реализован функционал регистрации FIDO2-аутентификатора для последующего его использования.
5. Доступ на страницу регистрации ключа должен быть запрещен для неавторизованных пользователей.
6. Если пользователь авторизован, то он не должен иметь доступ к странице входа в учетную запись.
7. Для авторизованных пользователей должна быть возможность выхода из учетной записи.
8. Приложение должно иметь возможность обращаться к клиентскому Android-приложению для nfc-подключения к электронному документу, проведения базового контроля доступа, пассивной и активной аутентификации, извлечения данных и проверки их соответствия требуемой сервером политике.

Помимо функциональных требований к приложению были выделены следующие требования к платформе, на которой ведется разработка:

1. Платформа должна поддерживать библиотеки, позволяющие серверу выступать в качестве доверенной стороны.
2. Платформа должна иметь возможности развертывания на Linux и контейнеризации, иметь необходимые компоненты.

Обоснование выбора платформ и средств разработки.

Фреймворк ASP.NET Core. В соответствии с разработанными требованиями в качестве платформы был выбран ASP.NET Core 8.0 – кроссплатформенный высокопроизводительный веб-фреймворк с открытым исходным кодом для создания современных приложений, подключенных к Интернету.

Фреймворк позволяет создавать как веб-API, в таком случае клиентская часть может быть реализована отдельно, так и приложение MVC, разрабатывая единое решение для пользовательского интерфейса и стороны сервера. Это удобно в данном случае, так как приложение должно поддерживать ввод и обработку запросов от пользователя, а также принимать и отправлять данные клиенту, реализуя спецификацию WebAuthn API.

ASP.NET Core содержит встроенный контейнер внедрения зависимостей, отвечающих за добавление того или иного функционала в веб-приложение. Эти зависимости также называются сервисами. В частности, для выполнения аутентификации фреймворк содержит сервис IAuthenticationService, который легко регистрируется на этапе конфигурации приложения с помощью метода builder.Services.AddAuthentication().

В контексте данной работы также важно, что ASP.NET Core предлагает встроенные средства авторизации и управления доступом, которые позволяют гибко настраивать политики безопасности. Во фреймворк встроены атрибуты для применения различных политик авторизации к отдельным контроллерам или действиям внутри контроллеров. Это дает возможность тонкой настройки требований к аутентификации и авторизации для каждой конечной точки приложения.

RSK FIDO2-компонент для ASP.NET Core. В качестве библиотеки, позволяющей разрабатываемому веб-приложению выступать в качестве проверяющей стороны, был выбран компонент, выпущенный компанией Rock Solid Knowledge (член альянса FIDO) для платформы ASP.NET Core 8.0[4].

Разработанная RSK FIDO2-библиотека позволяет веб-приложению выступать в роли проверяющей стороны WebAuthn, предоставляя сервис IFidoAuthentication, содержащий методы для формирования запросов к клиенту, который взаимодействует с WebAuthn API. Библиотека представляет собой компонент, предназначенный для промышленной разработки, и для его использования требуется предоставление компанией официальной лицензии[5].

Платформа Somee Web Hosting. Разработанное приложение было контейнеризовано (написаны соответствующие файлы Docker, Docker Compose) и развернуто на Somee Web Hosting.

Платформа позволяет загружать любое приложение и имеет возможность гибкой настройки серверной части. Somee позволяет упрощать и ускорять цикл разработки, снижает потребности в сложной работе с сервером, подходит для работы с нагруженными приложениями и быстрого масштабирования проектов.

Спецификация FIDO2 требует в процессе регистрации и аутентификации защищенное соединение между пользователем и клиентом. Поэтому для работы веб-приложения необходим SSL-сертификат – цифровой сертификат, удостоверяющий подлинность веб-сайта и позволяющий использовать зашифрованное соединение. Платформа Somee предоставляет возможности получения как бесплатного SSL-сертификата центра сертификации Let's Encrypt, так и сертификатов от сторонних поставщиков, таких как Comodo, Thawte или GeoTrust.

Взаимодействие с биометрическими документами, удостоверяющими личность. В биометрические документы встроена интегральная микросхема (чип), содержащая электронное средство биометрической идентификации с персональными данными владельца биометрического документа в соответствии с требованиями ICAO.

ICAO разработала LDS[6] – стандартизированную структуру данных, которая обеспечивает глобальную интероперабельность и семантическую совместимость. LDS содержит информацию о владельце электронного идентификатора и подразделяется на 16 групп данных. Каждая группа данных содержит определенные личные сведения, необходимые для идентификации личности. Это позволяет хранить и обмениваться информацией о личности в стандартизированном формате, обеспечивая эффективный обмен данными. Например, первая группа данных включает код документа, государство или организацию выдачи, имя владельца документа (имя владельца обычно представляется в двух частях – основной определитель и вторичный определитель), дату рождения, гражданство, пол, дату истечения срока действия документа, контрольные цифры; вторая группа данных содержит глобально интероперабельные биометрические параметры, необходимые для подтверждения личности с помощью eID: кодировку(и) биометрических характеристик лица, количество записанных кодировок, биометрический подтип, дату и время создания, срок действия и др.

Использование идентификации eID способствует совместимости на нескольких уровнях, что делает ее эффективным и удобным средством для подтверждения личности и обмена информацией в цифровой среде.

Для обеспечения высокого уровня защиты и целостности данных, представленных на электронном документе, стандартом ICAO предусмотрены следующие протоколы защиты[7]:

1. Базовый контроль доступа (BAC – Basic Access Control) предназначен для гарантии того, что доступ к данным карты возможен только при физическом доступе к ней. При считывании документа требуется ввести изображенные на карте номер, дату рождения владельца и срок

[4] RSK FIDO2 for ASP.NET documentation. – URL: https://www.identityserver.com/documentation/fido2/ (date of access: 12.01.2025).

[5] Компания Rock Solid Knowledge предоставила авторам статьи официальную лицензию на компонент RSK FIDO2 для ASP.NET Core.

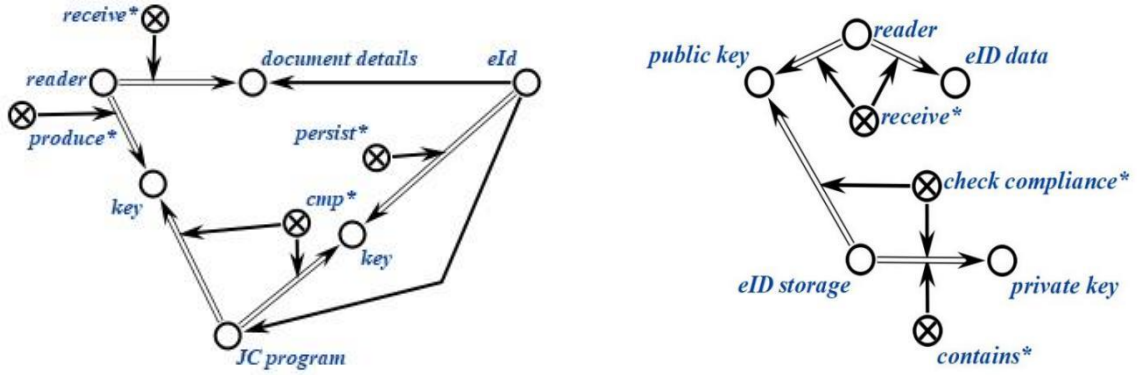
[6] Doc 9303 // ICAO. – URL: https://www.icao.int/publications/pages/publication.aspx?docnum=9303 (date of access: 12.01.2025).

[7] URL: https://www. icao.int/publications/pages/publication.aspx?docnum=9303

действия либо пин-код. На рис. 2, *a* изображена семантическая модель базового контроля доступа [7] с использованием спецификации SCg-кода [9].

2. Пассивная аутентификация позволяет считывающему устройству определять подлинность данных eID с помощью объекта защиты документа, проверяя подпись эмитента и хеш-значение полученных данных. Семантическая модель пассивной аутентификации представлена в работе [7].

3. Активная аутентификация, или аутентификация с чипом, предназначена для защиты данных документа от изменения и клонирования. Для этого при издании электронного документа создается пара открытого и закрытого ключей. Открытый ключ передается с данными считывателю, а закрытый помещается в защищенное хранилище документа. На рис. 2, *b* с использованием SCg-кода изображена семантическая модель активной аутентификации.



a) *b)*

Рис. 2. Модель базового контроля доступа (*a*); активной аутентификации (*b*)

Fig. 2. Basic access control model (a); active authentication model (b)

Для целей разработки и тестирования компонента системы, обеспечивающего взаимодействие с биометрическими документами, компанией ЗАО «АВЕСТ» – ведущим производителем систем защиты электронного документооборота в Республике Беларусь – был предоставлен полнофункциональный макет электронного идентификатора гражданина Республики Беларусь. Макет содержал необходимые тестовые данные о владельце и поддерживал описанные выше протоколы защиты.

Требования к мобильному компоненту системы беспарольного доступа. В качестве клиента для разрабатываемой системы было выбрано мобильное приложение. На основе открытости и широкого спектра компонентов и средств разработки была выбрана операционная система Android. Большинство современных android-устройств оснащены NFC-считывателями. Следовательно, от разрабатываемого приложения необходимо требовать поддержку обработки NFC-событий и извлечения из них данных.

Приложению следует корректно и достаточно быстро считывать и декодировать (расшифровывать) данные, находящиеся на электронном идентификаторе. Оно должно вызываться из браузера и иметь возможность взаимодействовать с сервером, выступающим в роли проверяющей стороны. Приложение должно использовать защищенное хранилище на android-устройстве. Передача атрибутов пользователя на сервер должна производиться без раскрытия содержимого этих атрибутов на основе использования протоколов и методов с нулевым разглашением информации. Приложение должно соответствовать стандартам ICAO, определяющим требования к электронным документам, их обработке, аутентификации и безопасности, обеспечивая надежность и совместимость с международными стандартами и регуляторными нормами. Оно должно осуществлять проверку целостности и подлинности электронных документов, проводить пассивную и активную аутентификацию с целью предотвращения подделки, изменения и несанкционированного доступа к информации.

Основываясь на вышеперечисленных требованиях, для реализации мобильного приложения был выбран язык программирования Kotlin с использованием дополнительных открытых библиотек для работы с электронными идентификаторами SCUBA[8], JMRTD[9] в среде разработки Android Studio (рис. 3).

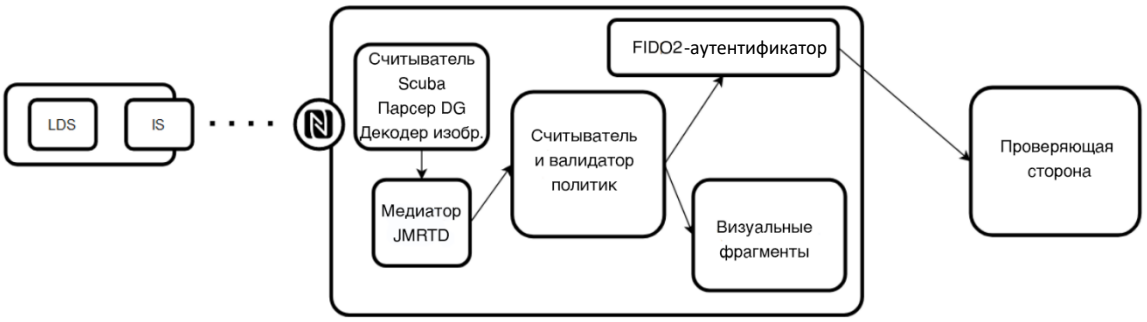


Рис. 3. Схема мобильного компонента системы

Fig. 3. Scheme of the mobile component of the system

Установка подключения к электронному идентификатору. Для прохождения базового контроля доступа приложению предоставляются номер электронного документа, срок его действия, дата рождения владельца.

Процесс извлечения данных из электронного документа состоит из нескольких шагов:

1. Приложение настраивается на обнаружение и реагирование на прикладывание электронного идентификатора к устройству.
2. После обнаружения eID приложение взаимодействует с NFC-чипом и считывает данные с использованием соответствующей технологии чтения NFC (в данном случае IsoDep).
3. После извлечения данных из электронного документа приложение выполняет базовый контроль доступа, сравнивая введенные пользователем данные (номер электронного документа, срок действия и дату рождения) с данными, полученными с электронного идентификатора. Если данные совпадают и проходят проверку, приложение может предоставить доступ к пользовательским атрибутам.
4. Выполняется пассивная и активная аутентификации.

Структура решения, полученного на основе интеграции eID-идентификации, FIDO2-аутентификации и атрибутивной авторизации. Структура решения с функциональной точки зрения изображена на рис. 4.

Для достижения модульности, гибкости и масштабируемости в приложении реализована трехуровневая архитектура (рис. 5). Она разделяет приложение на три основных уровня: представления, бизнес-логики и доступа к данным. Каждый уровень выполняет определенные функции и взаимодействует с другими уровнями для обеспечения эффективной работы всего приложения.

Для корректного взаимодействия между уровнями приложения и тестируемости использован встроенный в ASP.NET Core механизм внедрения зависимостей. При этом для каждой зависимости (например, сервиса или репозитория) определен интерфейс, а затем класс, который его реализует. На этапе конфигурирования приложения интерфейс и реализация регистрируются, после чего зависимость может быть получена из контейнера.

Для хранения учетных данных пользователей применяется СУБД MS SQL Server. Для взаимодействия веб-приложения с выбранной СУБД используется Entity Framework Core 6. EF Core оптимизирует взаимодействие с базой данных с помощью технологии «ленивой загрузки» (lazy loading) и отложенного выполнения запросов, что позволяет повышать производительность приложения, особенно при работе с большими объемами данных и высокой нагруженностью.

[8]SCUBA. – URL: https://scuba.sourceforge.net/ (date of access: 12.01.2025).
[9]JMRTD. – URL: https://jmrtd.org/ (date of access: 12.01.2025).

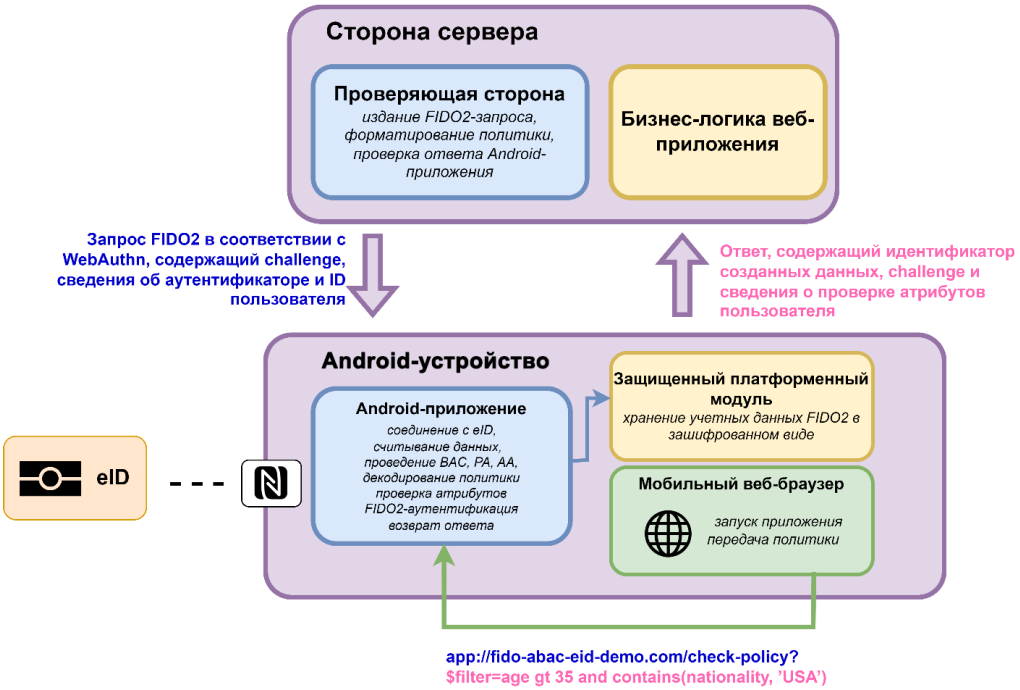


Рис. 4. Функциональная схема решения

Fig. 4. Functional scheme of the solution

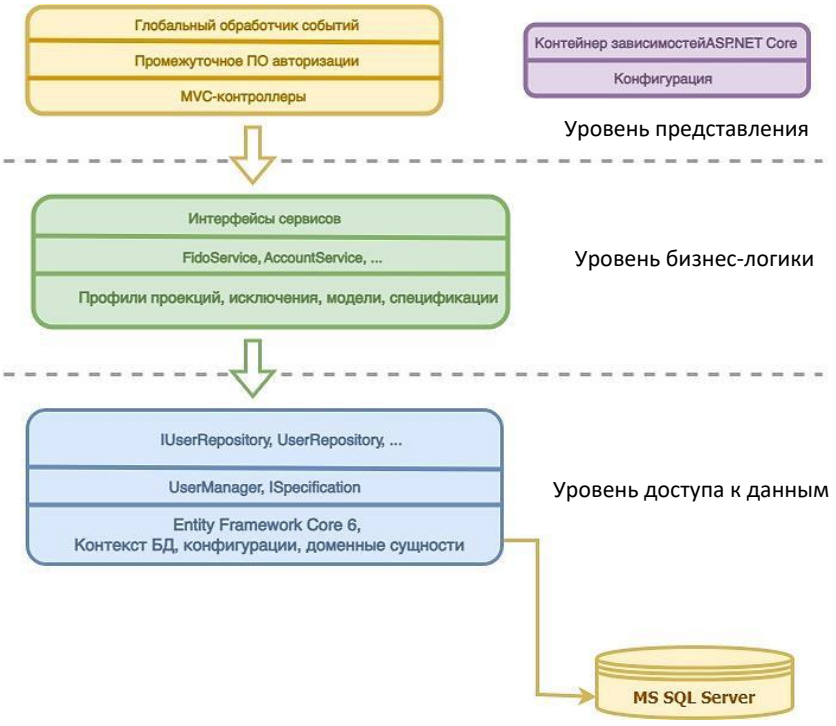


Рис. 5. Трехуровневая архитектура стороны сервера

Fig. 5. Three-tier server-side architecture

В базе данных не предусмотрено хранение чувствительных данных пользователя. Данные о годе рождения, месте жительства и др. будут получаться при считывании данных с электронного документа. Обрабатываться на стороне клиента и отправляться стороне сервера они не будут.

Работа с сущностями и контекстом базы данных осуществляется через класс UserManager, что повышает защищенность данных о пользователе, так как пароль хранится в виде хеш-значения, а в качестве «соли» применяется SecurityStamp. При этом использование UserManager инкапсулировано с помощью паттерна «Репозиторий», что делает доступ к данным из уровня бизнес-логики более простым и удобным. Сам объект класса UserManager внедряется в репозиторий через конструктор.

Интерфейс ISpecification<T> и метод расширения ApplySpecification<T> дают возможность осуществлять поиск пользователей по различным критериям.

Уровень бизнес-логики содержит интерфейсы и реализации сервисов для работы с пользовательскими учетными записями, аутентификации и реализации FIDO2. Одним из основных сервисов здесь является FidoService.

На уровне бизнес-логики осуществляется работа не с конкретной реализацией интерфейса UserRepository, обеспечивающего регистрацию пользователей, обновление данных, выборку пользователей, аутентификацию с использованием пароля и т. д, а с интерфейсом IUserRepository, что дает возможность при необходимости сменить хранилище данных, а также облегчает тестируемость.

Уровень представления. За FIDO-аутентификацию и аутентификацию с помощью логина и пароля отвечают соответственно два основных контроллера: FidoController и AccountController. AccountController содержит get-метод Login, который отправляет пользователю страницу с формой, в которую нужно ввести логин и пароль. Post-метод Login принимает данные из заполненной формы, идентифицирует пользователя (осуществляет поиск по логину) и сравнивает пароль с имеющимся. FidoController отвечает за формирование запросов к клиенту для создания новых учетных данных и аутентификации. При этом сам контроллер не содержит в себе бизнес-логику, а только отвечает за запуск нужных методов соответствующих сервисов.

Реализация процесса регистрации FIDO2-ключа. Для регистрации новых учетных данных в FidoController реализованы три метода: StartRegistration(), Register() и CompleteRegistration(). Регистрация ключа может быть выполнена только для авторизованных пользователей. Действия контроллера заключаются в том, чтобы сформировать запрос к клиенту на создание новых учетных данных и после получения ответа проверить его и зарегистрировать (рис. 6).

Рис. 6. Схема реализации процесса регистрации FIDO2-ключа

Fig. 6. Scheme of realization of the FIDO2-key registration process

Инициализация регистрации:

1. StartRegistration возвращает страницу с формой для ввода имени ключа.
2. Введенное имя передается в Register(), где идет обращение к сервису для работы с FIDO.

3. Генерируется challenge, создается объект запроса, содержащий данные о пользователе, проверяющей стороне и сам буфер challenge.

4. Запрос кодируется в base64 и передается в представление Register.

Обработка на клиенте:

1. JavaScript-скрипт на странице регистрации запускается при нажатии submit.
2. Декодируется challenge, формируются структуры:
   rp (проверяющая сторона) – указывает на домен сервера;
   user – содержит данные пользователя (userId, displayName, userHandle);
   pubKeyCredParams – указывает на приоритетные типы аутентификаторов (EC SHA-256, RSA SHA-256);
   authenticatorSelection – указывает на использование платформенных ключей;
   attestation – значение параметра (none) означает, что сервер не требует аттестации;
   timeout – указывает время на выполнение аутентификации.
3. Объект publicKey передается в функцию navigator.credentials.create (является частью WebAuthn API).

Аутентификация и отправка данных:

1. Аутентификатор запрашивает подтверждение пользователя.
2. После успешной верификации возвращается PublicKeyCredentials, содержащий id, rawId, response (с clientDataJSON и attestationObject).
3. Данные кодируются в base64 и отправляются серверу.

Завершение регистрации:

1. Контроллер Fido CompleteRegistration завершает этап регистрации.
2. Сервер проверяет и регистрирует учетные данные.
3. Данные хранятся в InMemoryKeyStore в памяти сервера, возможно подключение внешней базы данных.

Семантическая модель регистрации FIDO2-ключа была представлена авторами ранее в работе [10].

Реализация процесса аутентификации пользователя с FIDO2-ключом. На стороне сервера процесс аутентификации пользователя в контроллере Fido представлен двумя методами: Login и CompleteLogin. Действия контроллера заключаются в том, чтобы проверить, существуют ли требуемые учетные данные в хранилище, и сформировать запрос к клиенту. После этого, получив ответ, содержащий подпись, проверить его и подтвердить аутентификацию пользователя.

Разработанное приложение было развернуто на сервисе Somee, также был установлен SSL-сертификат, предоставляемый платформой. Его работоспособность была проверена с помощью устройств на Windows, Android и IOS. При этом использовались браузеры Chrome, Opera, Edge, Mozilla и Safari.

Передача политики клиентской стороне. Важным вопросом является формат, в котором политика будет передаваться от сервера клиенту. Выбранный формат должен поддерживать отношения «больше», «меньше», «равно», например «возраст больше 25», а также конъюнкцию и дизъюнкцию условий. В качестве такого формата был выбран OData[10]. OData (Open Data Protocol) – открытый протокол, определяющий стандартные способы запроса и изменения данных через REST API, а также стандартный формат для запросов и ответов, который основан на широко используемых протоколах и форматах данных. Формат запроса OData можно использовать для передачи политики конечной точки, к которой пользователь запрашивает доступ, приложению, взаимодействующему с электронным документом.

Таким образом, для того чтобы на странице веб-приложения открыть Android-приложение для взаимодействия с электронным документом, а также передать необходимую политику, JavaScript-код страницы должен направить запрос по адресу вида *app://fido-abac-eid-demo.com/check-policy?$filter=age gt 35 and contains(nationality, 'USA').*

[10]Документация. – URL: https://www.odata.org/documentation/ (дата обращения: 12.01.2025).

Конфигурация разработанного приложения. После добавления библиотеки Rsk.AspNetCore.Fido нужно подключить соответствующий сервис с помощью метода builder.Services.AddFido(), в который требуется передать лицензионный ключ и тип лицензии.

Лицензионный ключ является конфиденциальной информацией и, следовательно, требует более безопасного хранения, чем стандартный файл appsettings.json, являющийся частью большинства ASP.NETCore-приложений. При развертывании на предприятии распространенным решением является использование хранилища Azure Key Vault, созданного для этих целей. На этапе же разработки целесообразно использовать хранилище секретов Windows (User Secrets).

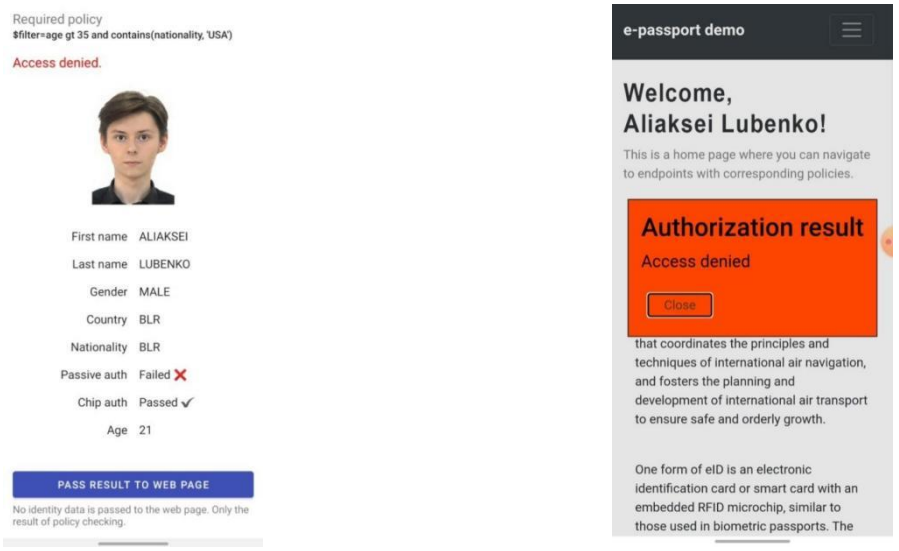
Для возможности аутентификации с помощью логина и пароля следует подключить аутентификацию на основе Cookie. Для этого вызывается метод builder.Services.AddAuthentication(), с помощью параметра устанавливается использование схемы аутентификации на основе Cookie. Также на данном этапе указывается, что приложение будет использовать MVC.

Демонстрация разработанного веб-приложения. Рассмотрим основные элементы полученной реализации. Веб-приложение представляет собой набор конечных точек (страниц и ресурсов), также для удобства навигации реализованы навигационная панель и подвал.

После парольной аутентификации пользователь переходит на домашнюю (тестовую) страницу, содержащую информацию об электронных идентификаторах. На тестовой странице размещены ссылки, ведущие к двум защищенным ресурсам, доступ к которым ограничен разными политиками. Первый требует от пользователя возраст более 18 лет или гражданство Республики Беларусь. Для доступа ко второму необходимо иметь возраст более 35 лет и гражданство Соединенных Штатов Америки.

При переходе по каждой из ссылок запускается Android-приложение, в которое передается политика соответствующего ресурса. Приложение проверяет подлинность личности и соответствие пользовательских атрибутов требуемой политике. Результат проверки возвращается на веб-ресурс.

На рис. 7 изображен снимок экрана мобильного приложения после считывания данных с тестовой eID-карты, установки личности с использованием FIDO2-аутентификации и проверки атрибутов пользователя, а также страница в веб-приложении с результатом авторизации.



a) *b)*

Рис. 7. Результат проверки подлинности личности и соответствия пользовательских атрибутов требуемой политике доступа к ресурсу на основании тестовой eID-карты Республики Беларусь: *a)* экран мобильного устройства с NFC-считывателем; *b)* веб-приложение

Fig. 7. Result of identity authentication and compliance of user attributes with the required resource access policy based on the test eID-card of the Republic of Belarus: a) mobile device screen with NFC reader; b) web application

Можно заметить, что пользователю отказано в доступе. В первую очередь он не соответствует требуемой политике (указана в верхней части снимка экрана). Также не была пройдена пассивная аутентификация. Дело в том, что на тестовой eID-карте с данными одного из разработчиков отсутствует подпись издателя (если бы она была, это был бы настоящий паспорт). Поэтому при работе с тестовой картой была исключена пассивная аутентификация.

Пример с прохождением пассивной аутентификации на основании подлинной eID-карты Республики Беларусь здесь не приводится из-за необходимости защиты персональных данных.

Заключение. В настоящей публикации авторы представили исследование по разработке и первоначальной оценке прототипа системы управления доступом к информационным ресурсам путем аутентификации по спецификации FIDO2 и модели управления доступом на основе атрибутов. При этом в качестве источника пользовательских атрибутов применяются средства электронной идентификации, удовлетворяющие стандартам Международной организации гражданской авиации, в том числе eID-карта Республики Беларусь.

Разработанное приложение было контейнеризовано и развернуто на онлайн-сервере, его работоспособность проверена с различных платформ с помощью распространенных браузеров.

В форме SCn- и SCg-кода технологии OSTIS представлены основные понятия и концепции, связанные с FIDO2-аутентификацией, предложена семантическая сеть, описывающая взаимодействие сервера, клиента и аутентификатора при регистрации WebAuthn-ключа и аутентификации. Это позволяет обеспечить гибкий и масштабируемый подход к аутентификации на основе онтологии.

Показано, что разработанная система безопасности способствует совместимости с информационными системами на нескольких уровнях, в том числе техническом и семантическом. Предполагается, что построенная система должна обладать гибкостью, масштабируемостью и интероперабельностью вследствие поддержки стандартов ICAO и будет адаптирована к использованию в корпоративных и муниципальных системах.

Вклад авторов. *А. А. Жидович* внес существенный вклад в реализацию работы и написание текста статьи; *А. А. Лубенько* реализовал мобильный компонент системы беспарольного доступа, участвовал в написании текста; *И. С. Войтешенко* обосновал концепцию и актуальность статьи, осуществил анализ и интерпретацию результатов работы, проанализировал содержание текста, утвердил окончательный вариант статьи.

Список использованных источников

1. Angelogianni, A. How many FIDO protocols are needed? Analysing the technology, security and compliance / A. Angelogianni, I. Politis, C. Xenakis // ACM Computing Surveys. – 2024. – Vol. 56, iss. 8. – P. 1–51. – DOI: 10.1145/3654661.

2. Challenges with Passwordless FIDO2 in an Enterprise Setting: A Usability Study / M. Kepkowski, M. MachulakIan, I. Wood, D. Kaafar. – URL: https://arxiv.org/abs/2308.08096 (date of access: 08.11.2024).

3. FIDO2 passwordless authentication for remote devices / S. Dixit, A. Gupta, R. Jain [et al.] // Networks and Systems in Cybernetics : Proc. of 12th Computer Science On-line Conf. 2023. – Springer, 2023. – Vol. 2. – P. 349–362. – DOI: 10.1007/978-3-031-35317-8_32.

4. Hoefling, D. Understanding How FIDO Makes Passwordless Authentication Possible / D. Hoefling. – URL: https://practical365.com/understanding-how-fido-makes-passwordless authentication-possible/ (date of access: 27.11.2024).

5. Fast IDentity online with anonymous credentials (FIDO-AC) / W.-Z. Yeoh, M. Kepkowski, G. Heide [et al.] // Proc. of the 32nd USENIX Security Symp., Anaheim, CA, USA, 9–11 Aug. 2023. – Anaheim, 2023. – P. 3029–3046.

6. Brodsky, Z. Using MITM to bypass FIDO2 phishing-resistant protection / Z. Brodsky. – URL: https://www.silverfort.com/blog/using-mitm-to-bypass-fido2/ (date of access: 08.11.2024).

7. Zhidovich, A. Semantic notation of access control technology based on eID identification, FIDO2-authentication and attribute-based authorization in digital environment / A. Zhidovich, A. Lubenko, I. Vojteshenko // Open Semantic Technologies for Intelligent Systems. – 2024. – No. 8. – P. 371–376.

8. Garoffolo, A. Zendoo: A ZK-SNARK verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains / A. Garoffolo, D. Kaidalov, R. Oliynykov // 2020 IEEE 40th Intern. Conf. on Distributed Computing Systems (ICDCS), Singapore, 29 Nov. – 01 Dec. 2020. – Singapore, 2020. – P. 1257–1262.

9. Технология комплексной поддержки жизненного цикла семантически совместимых интеллектуальных компьютерных систем нового поколения / под общ. ред. В. В. Голенкова. – Минск : Бестпринт, 2023. – 1064 с.

10. Semantic approach to designing applications with passwordless authentication according to the FIDO2 specification / A. Zhidovich, A. Lubenko, I. Vojteshenko, A. Andrushevich // Open Semantic Technologies for Intelligent Systems. – 2023. – No. 7. – P. 311–316.

References

1. Angelogianni A., Politis I., Xenakis C. How many FIDO protocols are needed? Analysing the technology, security and compliance. *ACM Computing Surveys*, 2024, vol. 56, iss. 8, pp. 1–51. DOI: 10.1145/3654661.

2. Kepkowski M., MachulakIan M., Wood I., Kaafar D. Challenges with Passwordless FIDO2 in an Enterprise Setting: A Usability Study. Available at: https://arxiv.org/abs/2308.08096 (accessed 08.11.2024).

3. Dixit S., Gupta A., Jain R., Joshi R., Gonge S., Kotecha K. FIDO2 passwordless authentication for remote devices. *Networks and Systems in Cybernetics: Proceedings of 12th Computer Science On-line Conference 2023*. Springer, 2023, vol. 2, pp. 349–362. DOI: 10.1007/978-3-031-35317-8_32.

4. Hoefling D. Understanding How FIDO Makes Passwordless Authentication Possible. Available at: https://practical365.com/understanding-how-fido-makes-passwordless authentication-possible/ (accessed 27.11.2024).

5. Yeoh W.-Z., Kepkowski M., Heide G., Kaafar D., Hanzlik L. Fast IDentity online with anonymous credentials (FIDO-AC). *Proceedings of the 32nd USENIX Security Symposium, Anaheim, CA, USA, 9–11 Aug. 2023*. Anaheim, 2023, pp. 3029–3046.

6. Brodsky Z. Using MITM to bypass FIDO2 phishing-resistant protection. Available at: https://www.silverfort.com/blog/using-mitm-to-bypass-fido2/ (accessed 08.11.2024).

7. Zhidovich A., Lubenko A., Vojteshenko I. Semantic notation of access control technology based on eID identification, FIDO2-authentication and attribute-based authorization in digital environment. *Open Semantic Technologies for Intelligent Systems*, 2024, no. 8, pp. 371–376.

8. Garoffolo A., Kaidalov D., Oliynykov R. Zendoo: A ZK-SNARK verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains. *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November – 01 December 2020*. Singapore, 2020, pp. 1257–1262.

9. Golenkov V. V. (ed.). Tehnologija kompleksnoj podderzhki zhiznennogo cikla semanticheski sovmestimyh intellektual'nyh komp'juternyh sistem novogo pokolenija. *Technology for Integrated Life Cycle Support of Semantically Compatible Intelligent Computer Systems of the New Generation*. Minsk, Bestprint, 2023, 1064 p.

10. Zhidovich A., Lubenko A., Vojteshenko I., Andrushevich A. Semantic approach to designing applications with passwordless authentication according to the FIDO2 specification. *Open Semantic Technologies for Intelligent Systems*, 2023, no. 7, pp. 311–316.

Информация об авторах

Жидович Антон Андреевич, выпускник кафедры технологий программирования факультета прикладной математики и информатики, Белорусский государственный университет.
E-mail: anton.zhidovich@gmail.com

Лубенько Алексей Анатольевич, выпускник кафедры технологий программирования факультета прикладной математики и информатики, Белорусский государственный университет.
E-mail: alexeilubenko02@gmail.com

Войтешенко Иосиф Станиславович, кандидат технических наук, доцент, доцент кафедры технологий программирования факультета прикладной математики и информатики, Белорусский государственный университет.
E-mail: voit@bsu.by
https://orcid.org/0000-0002-0134-1793

Information about the authors

Anton A. Zhidovich, Graduate of the Department of Programming Technologies, Faculty of Applied Mathematics and Informatics, Belarusian State University.
E-mail: anton.zhidovich@gmail.com

Alexei A. Lubenko, Graduate of the Department of Programming Technologies, Faculty of Applied Mathematics and Informatics, Belarusian State University.
E-mail: alexeilubenko02@gmail.com

Iosif S. Vojteshenko, Ph. D. (Eng.), Assoc. Prof., Assoc. Prof. of the Department of Programming Technologies of the Faculty of Applied Mathematics and Informatics, Belarusian State University.
E-mail: voit@bsu.by
https://orcid.org/0000-0002-0134-1793