Научная статья
УДК 681.3; DOI: 10.61260/2304-0130-2025-2-56-60

ИДЕНТИФИКАЦИЯ КАК СРЕДСТВО ДОСТУПА К ИНФОРМАЦИИ

✉**Лабинский Александр Юрьевич.**
Санкт-Петербургский университет ГПС МЧС России, Санкт-Петербург, Россия
✉***labinskyi.a@igps.ru***

Аннотация. Рассмотрены особенности и недостатки традиционного способа идентификации пользователя в сети и особенности и преимущества идентификации с помощью одноразового пароля, полученного с помощью токена безопасности. Рассмотрены особенности аппаратных и программных токенов.

Подробно рассмотрена идентификация пользователя в сети для случая использования программного токена, включая установку приложения на мобильное устройство пользователя, регистрацию устройства пользователя на сервере сети и ввод пользователем PIN-кода в приложении программного токена. В результате устройство пользователя генерирует одноразовый пароль, используемый для входа в систему сети.

Отмечены преимущества использования программных токенов.

Для программных токенов подробно рассмотрены алгоритмы генерации одноразовых кодов, включая блок-схему и текст укрупненных модулей на языке C#. Подробно рассмотрены шаги алгоритма генерации одноразового пароля.

Ключевые слова: безопасность информации, безопасный доступ, идентификация пользователя, компьютерные сети, аппаратный токен безопасности, программный токен безопасности, хэш-функция, алгоритм шифрования, одноразовый пароль

Для цитирования: Лабинский А.Ю. Идентификация как средство доступа к информации // Надзорная деятельность и судебная экспертиза в системе безопасности. 2025. № 2. С. 56–60. DOI: 10.61260/2304-0130-2025-2-56-60.

Введение

Безопасность информации, хранящейся в компьютерной сети, обеспечивается набором требований и политик, предъявляемых к инфраструктуре сети с целью анализа её работы и обеспечения безопасного доступа к информации.

Традиционный способ идентификации в компьютерной сети представляет собой ввод логина и пароля. Их эталонные значения хранятся в специальной базе данных сети. Однако такой способ идентификации в сети имеет как минимум два существенных недостатка: пароль может быть забыт пользователем или перехвачен, подсмотрен, подобран злоумышленником.

Поэтому в настоящее время для обеспечения безопасного доступа применяются различные аппаратные и программные средства, среди которых широко используются токены. Они являются средством идентификации пользователя или отдельного сеанса работы в компьютерной сети.

Аппаратный токен представляет собой компактное устройство, которое обеспечивает электронное удостоверение личности пользователя, а также удаленный доступ к информационным ресурсам сети. Он имеет небольшие размеры, может хранить криптографические ключи в виде электронной подписи или биометрических данных, обычно имеет USB-разъемы или интерфейс Bluetooth для передачи заданной последовательности ключевых символов в систему доступа сети.

Аппаратные токены могут содержать фиксированный пароль доступа для каждого сеанса аутентификации, одноразовый сгенерированный пароль с определенным интервалом времени, одноразовый асинхронный пароль или пароль, созданный с помощью криптографии с открытым ключом.

Программный токен имеет систему авторизации, которая привязывается к конкретному клиенту сети, сеансу работы в сети или пакету данных, размещенных в сети. Система авторизации программного токена обеспечивает генерацию зашифрованной последовательности символов для точной идентификации клиента сети и определения уровня его привилегий.

Сформулируем постановку задачи. Нужно рассмотреть особенности обеспечения безопасного доступа к информации, размещаемой в компьютерных сетях, с помощью токенов безопасности. Тема статьи актуальна, так как количество сетевых атак и ущерб от несанкционированного доступа к информации постоянно увеличиваются. Поэтому средствам обеспечения безопасного доступа к информации посвящено много работ [1–8].

Особенности программных токенов

Программные токены обеспечивают ограниченность по времени действия пароля клиента сети. По истечении этого времени необходимо менять пароль. Таким образом, пароль действителен для одного входа в систему сети. При каждом следующем запросе доступа в сеть необходим новый пароль.

Идентификация пользователя в сети происходит следующим образом. На устройство пользователя устанавливается приложение в виде программного (виртуального) токена. Затем пользователь регистрирует своё устройство на сервере сети. После ввода пользователем PIN-кода в приложении программного токена устройство пользователя генерирует одноразовый пароль, используемый для входа в систему сети.

Приложение программного токена создает одноразовый пароль, используя следующую информацию: текущее время с точностью до десятой доли секунды, четырехзначный PIN-код и шестнадцатеричный код, созданный при регистрации устройства в системе. Далее приложение программного токена производит обработку данной информации с использованием алгоритма шифрования MD5, который из входной информации создает 128-битовое хэш-значение.

Преимущества использования программных токенов заключатся в следующем:

– аппаратный токен может быть заменен любым мобильным устройством, например, мобильным телефоном;

– программный токен работает на всех мобильный устройствах, поддерживающих java-приложения (iPhone, Nokia, Siemens, Motorola, Sony и т.п.);

– получаемый одноразовый пароль действует в течение одного сеанса работы в сети.

Алгоритмы генерации одноразовых кодов

В основе алгоритмов генерации одноразовых кодов находится код идентификации сообщений, использующий хэш-функции (HMAC – Hash-based Message Authentification Code).

Алгоритм генерации одноразового пароля содержит следующие шаги:

– создание 20-байтовой строки на основе входной информации с использованием хэш-функции;

– извлечение 4 байтов информации из 20-байтовой строки;

– преобразование извлеченного значения в число, которое делится на 10 в степени N, где N – количество символов в одноразовом пароле;

– использование остатка от деления в качестве одноразового пароля.

Блок-схема алгоритма генерации одноразового пароля представлена на рисунке.

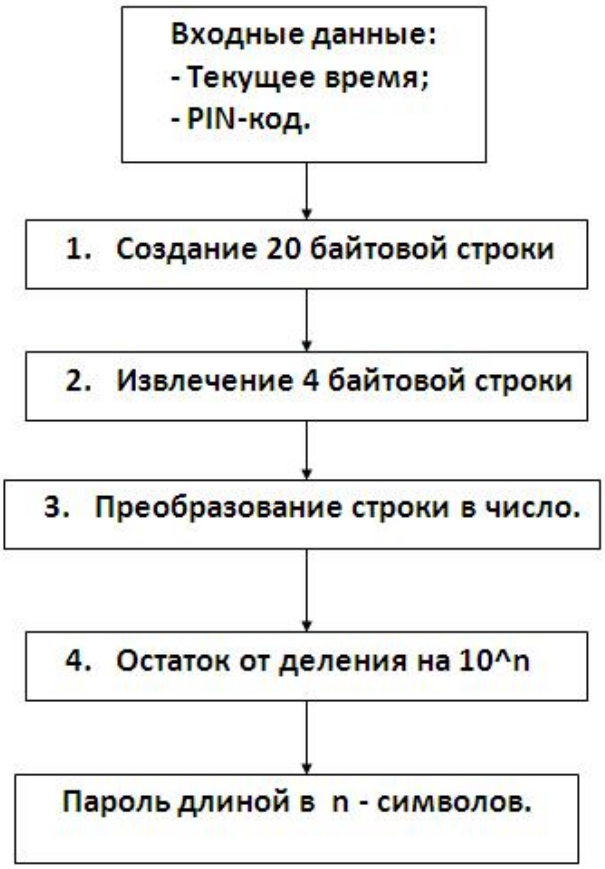

Рис. **Блок-схема алгоритма генерации пароля**

Укрупненный алгоритм программы на языке C# в виде отдельных модулей представлен в табл. 1–5.

Таблица 1

Укрупненный текст модуля на языке C#

«const Crypto = window.crypto.subtle;
const encoder = new TextEncoder('utf-8');
const secretBytes = encoder.encode(secret);
const key = await Crypto.importKey('raw',
secretBytes,
{ name: 'HMAC', hash: { name: 'SHA-1' } },
false, ['sign']);»

Таблица 2

Укрупненный текст модуля на языке C#

« function padCounter(counter)
{ const buffer = new ArrayBuffer(8);
const bView = new DataView(buffer);
const byteString = '0'.repeat(64); // 8 bytes
const bCounter = (byteString + counter.toString(2)).slice(-64);
for (let byte = 0; byte < 64; byte += 8)
{ const byteValue = parseInt(bCounter.slice(byte, byte + 8), 2);
bView.setUint8(byte / 8, byteValue);
} return buffer;}»

Таблица 3

Укрупненный текст модуля на языке C#
« function DT(HS) { const offset = HS[19] & 0b1111; const P = ((HS[offset] & 0x7f) << 24) | (HS[offset + 1] << 16) | (HS[offset + 2] << 8) | HS[offset + 3] const pString = P.toString(2); return pString;}»

Таблица 4

Укрупненный текст модуля на языке C#
« function truncate(uKey) { const Sbits = DT(uKey); const Snum = parseInt(Sbits, 2); return Snum; }»

Таблица 5

Укрупненный текст модуля на языке C#

«async function generateHOTP(secret, counter)
{ const key = await generateKey(secret,
counter);
const uKey = new Uint8Array(key);
const Snum = truncate(uKey);
const padded = ('000000' + (Snum % (10
** 6))).slice(-6);
return padded; }»

В табл. 1 представлен модуль переменных и констант, в табл. 2 – модуль создания 20-байтовой строки, в табл. 3 – модуль извлечения 4-байтовой строки, в табл. 4 – модуль преобразования строки в число, в табл. 5 – модуль получения остатка от деления.

Вывод

Рассмотрены особенности аппаратных и программных токенов безопасности. Для программных токенов рассмотрены алгоритмы генерации одноразовых кодов, включая блок-схему и текст укрупненных модулей на языке C#.

Таким образом, идентификация с использованием создаваемых программным токеном временных паролей является в настоящее время самым надежным способом идентификации пользователя сети. Использование данного способа идентификации позволяет устранить риски, возникающие при использовании стандартной парольной идентификации, и в результате надежно защитить данные, размещаемые в компьютерной сети.

Список источников

1. ГОСТ Р 51275–2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения // ЭЛЕКТРОННЫЙ ФОНД правовой и нормативно-технической документации. URL: http://www.docs.cntd.ru (дата обращения: 24.03.2025).

2. Ананченко И.В., Мусаев А.А. Защита приложений. М.: Труды СПИИРАН, 2013.

3. Концепция токена безопасности для устройств программного управления / А.Н. Кокоулин [и др.] // Технические науки – от теории к практике». 2016. № 1.

4. Маркова С.В. Выявление уязвимостей в информационных системах // Фундаментальные исследования. 2022. № 9.

5. Шелупанов А. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. М.: Изд-во «Горячая линия – телеком», 2009.

6. Johnson M. New adwanced personal data protection. Wiley Information Technologies, 2016.

7. Joseph, Migga, Kizza. Computer Network Security. Springer Science & Business Media, 2005.

8. Jie Wang; Zachary A. Kissel. Introduction to Network Security: Theory and Practice. Wiley, 2015.

9. Owen Poole. Network Security. Routledge, 2007.

10. USB-ключи и смарт-карты eToken. URL: http://www.aladdin-rd.ru/catalog/etoken (дата обращения: 24.03.2025).

Информация о статье: статья поступила в редакцию: 16.04.2025; принята к публикации: 29.05.2025

Информация об авторах:

Лабинский Александр Юрьевич, доцент кафедры прикладной математики и информационных технологий Санкт-Петербургского университета ГПС МЧС России (196105, Санкт-Петербург, Московский пр., д. 149), кандидат технических наук, доцент, e-mail: labinskyi.a@igps.ru, https://orcid.org/0000-0001-2735-4189, SPIN-код: 8338-4230