**УДК 004.8:004.42**
**DOI https://doi.org/10.32689/maup.it.2024.4.6**

***Олександр ГОРДІЄНКО***
*кандидат технічних наук, доцент кафедри комп'ютерних інформаційних систем та технологій, Інститут комп'ютерно-інформаційних технологій та дизайну ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом», oleksandr.m.hordiienko@gmail.com*
***ORCID:** 0009-0002-7764-8668*

***Аліна КОВАЛЬ***
*викладач кафедри комп'ютерних інформаційних систем та технологій*
*Інститут комп'ютерно-інформаційних технологій та дизайну*
*ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом», cora9393@gmail.com*
***ORCID:** 0009-0001-7379-5065*

## ВИКОРИСТАННЯ КРИПТОГРАФІЇ ЯК СЕРВІСУ У ВЕБ ПРОГРАМУВАННІ

***Анотація.*** *Стаття присвячена дослідженню використання криптографії як сервісу (Cryptography as a Service, CaaS) у веб-програмуванні, що є важливою складовою сучасних підходів до забезпечення безпеки веб-додатків. Зважаючи на постійний ріст кількості онлайн-сервісів і зростаючу важливість захисту персональних даних, необхідність у надійних методах криптографії є незаперечною. Веб-розробники часто стикаються з проблемою інтеграції криптографічних функцій у свої додатки, що потребує значних витрат часу, зусиль та ресурсів. Криптографія як сервіс надає зручну і ефективну альтернативу, дозволяючи швидко інтегрувати захист даних через API, без необхідності глибоких знань у сфері криптографії.*

***Мета статті.*** *Дослідити концепцію використання криптографії як сервісу CaaS, Cryptography у веб-програмуванні, визначити її переваги, обмеження та перспективи для забезпечення безпеки веб-додатків, а також розробити рекомендації щодо інтеграції CaaS у сучасні веб-розробницькі процеси.*

***Методологія.*** *Проведено огляд існуючих сервісів CaaS (наприклад, AWS Key Management Service, Azure Key Vault). Проаналізовано технічні аспекти інтеграції криптографічних сервісів у веб-додатки на прикладах популярних мов програмування (JavaScript, Python, Java). Виконано порівняльний аналіз ефективності та безпеки CaaS у порівнянні з традиційними методами реалізації криптографії в веб-програмуванні. Проведено тестування практичного використання CaaS у моделюванні сценаріїв для шифрування даних, управління ключами та автентифікації.*

***Наукова новизна.*** *Представлено систематизований аналіз можливостей CaaS у веб-програмуванні, зокрема для шифрування даних, цифрового підпису, автентифікації та управління ключами. Описано нові підходи до зниження навантаження на розробників шляхом делегування складних криптографічних операцій хмарним сервісам. Запропоновано рекомендації щодо вибору та інтеграції CaaS з урахуванням специфіки веб-додатків.*

***Висновок.*** *Криптографія як сервіс пропонує розробникам ефективні інструменти для підвищення безпеки веб-додатків, дозволяючи делегувати складні криптографічні завдання спеціалізованим платформам. Це спрощує впровадження надійних механізмів шифрування, автентифікації та управління ключами, що є критично важливими для сучасних веб-систем. Однак для ефективного використання CaaS необхідно враховувати специфіку додатків, забезпечувати відповідність нормативним вимогам і мінімізувати залежність від зовнішніх сервісів шляхом впровадження резервних механізмів.*

*Загалом, стаття пропонує комплексний огляд криптографії як сервісу в контексті веб-програмування та демонструє, як цей підхід може значно полегшити розробку безпечних веб-додатків, зберігаючи при цьому високу ефективність і надійність системи.*

***Ключові слова:*** *Цифровий підпис, PKI (Public Key Infrastructure), Digital Certificates, CA, Certification Authority, (RA, Registration Authority), (DB, Database), (CMS, Certificate Management System), (CRL, Certificate Revocation List), (Public and Private Keys), ECDSA, RSA, SHA-2.*

## Oleksandr HORDIIENKO, Alina KOVAL. USING CRYPTOGRAPHY AS A SERVICE IN WEB PROGRAMMING

***Abstract.*** *The article is devoted to the study of the use of Cryptography as a Service (CaaS) in web programming, which is an important component of modern approaches to ensuring the security of web applications. Given the constant growth of the number of online services and the growing importance of protecting personal data, the need for reliable cryptography methods is undeniable. Web developers often face the problem of integrating cryptographic functions into their applications, which requires significant expenditure of time, effort and resources. Cryptography as a service provides a convenient and effective alternative, allowing you to quickly integrate data protection via API, without the need for in-depth knowledge in the field of cryptography.*

***The purpose of the article.*** *To investigate the concept of using cryptography as a CaaS service, Cryptography in web programming, to determine its advantages, limitations and prospects for ensuring the security of web applications, as well as to develop recommendations for integrating CaaS into modern web development processes.*

***Methodology.*** *A review of existing CaaS services (for example, AWS Key Management Service, Azure Key Vault) was conducted. The technical aspects of integrating cryptographic services into web applications were analyzed using examples of popular programming languages (JavaScript, Python, Java). A comparative analysis of the effectiveness and security of CaaS was performed in comparison with traditional methods of implementing cryptography in web programming. The practical use of CaaS was tested in modeling scenarios for data encryption, key management and authentication.*

***Scientific novelty.*** *A systematic analysis of the capabilities of CaaS in web programming, in particular for data encryption, digital signature, authentication and key management, is presented. New approaches to reducing the burden on developers by delegating complex cryptographic operations to cloud services are described. Recommendations for the selection and integration of CaaS are proposed, taking into account the specifics of web applications.*

***Conclusion.*** *Cryptography as a service offers developers effective tools for improving the security of web applications, allowing them to delegate complex cryptographic tasks to specialized platforms. This simplifies the implementation of reliable encryption, authentication, and key management mechanisms, which are critical for modern web systems. However, for effective use of CaaS, it is necessary to take into account the specifics of applications, ensure compliance with regulatory requirements, and minimize dependence on external services by implementing backup mechanisms.*

*Overall, the article offers a comprehensive overview of cryptography as a service in the context of web programming and demonstrates how this approach can significantly facilitate the development of secure web applications, while maintaining high system performance and reliability.*

***Key words:*** *Digital Signature, PKI (Public Key Infrastructure), Digital Certificates, CA, Certification Authority, (RA, Registration Authority), (DB, Database), (CMS, Certificate Management System), (CRL, Certificate Revocation List), (Public and Private Keys), ECDSA, RSA, SHA-2.*

**Вступ. Цифровий підпис** – це електронний аналог власноручного підпису, який використовується для автентифікації та забезпечення цілісності електронних документів. Він дозволяє підтвердити, що документ не було змінено після підписання, а також, що підписант є автором цього документа.

Цифровий підпис базується на криптографії з відкритим ключем і використовує пару ключів: приватний і публічний [1-3, 7, 12, 13].

Криптографія з відкритим ключем.

PKI (Public Key Infrastructure) – це система, яка використовує криптографію з відкритим ключем для забезпечення безпеки в електронному середовищі, а також для управління та обміну електронними підписами, сертифікатами та іншими безпечними елементами. PKI дозволяє здійснювати аутентифікацію, шифрування та забезпечення цілісності даних, використовуючи пару ключів: публічний і приватний [1, 2].

Основні компоненти PKI:

1. *Сертифікати (Digital Certificates)* – це електронні документи, які містять публічний ключ і інформацію про власника цього ключа (наприклад, ім'я, організація, термін дії тощо). Сертифікати підписуються центром сертифікації (CA), що гарантує їх достовірність. Вони використовуються для підтвердження ідентичності суб'єктів і для забезпечення довіри до публічного ключа.

2. *Центр сертифікації (CA, Certification Authority)* – це організація або служба, яка видає сертифікати та перевіряє особу або організацію перед видачею сертифікату. CA підписує сертифікати, підтверджуючи їх дійсність і зв'язок з певною особою чи організацією. Прикладом CA може бути компанія, що надає послуги цифрових підписів (наприклад, VeriSign, DigiCert або Національний центр сертифікації ключів в Україні).

3. *Реєстраційний орган (RA, Registration Authority)*. RA працює в тісній взаємодії з центром сертифікації. Це орган або служба, яка відповідає за прийом запитів на сертифікати, перевірку особи та передачу запитів в CA для видачі сертифікатів. RA може здійснювати перевірку особи за допомогою документів, особистої перевірки тощо.

4. *Реєстраційна база даних (DB, Database)* – це база даних, де зберігаються сертифікати, статуси сертифікатів, ключі та інша важлива інформація для підтримки PKI-системи.

5. *Система управління сертифікатами (CMS, Certificate Management System).* CMS використовує сертифікати та допомагає у їх створенні, обміні, оновленні, відкликанні тощо.

6. *Система відкликання сертифікатів (CRL, Certificate Revocation List)* –це список сертифікатів, які були відкликані до завершення терміну їх дії. Відкликання сертифікатів може відбутися з різних причин: наприклад, якщо приватний ключ був скомпрометований, чи користувач припинив свою діяльність. CRL допомагає визначити, чи є сертифікат недійсним.

7. *Пара публічного і приватного ключів (Public and Private Keys)*. Приватний ключ використовується для підписання документів і повинен бути захищений. Лише підписант має доступ до свого приватного ключа. Публічний ключ використовується для перевірки підпису. Він може бути доступний будь-кому, хто хоче перевірити достовірність підписаного документа.

**Функцій PKI для безпеки даних у цифровому середовищі** [1].

1. *Аутентифікація*. Завдяки використанню сертифікатів, PKI дозволяє перевіряти, що обидві сторони (наприклад, підписувач та отримувач) є тими, за кого вони себе видають. Це важливо для онлайн-транзакцій, електронних підписів та інших операцій, що вимагають верифікації особи.

2. *Шифрування*. Для шифрування даних використовується публічний ключ: будь-хто може зашифрувати інформацію за допомогою публічного ключа, але тільки власник відповідного приватного ключа може її дешифрувати.

3. *Цілісність і підписання*. За допомогою криптографічних підписів з використанням приватного ключа можна гарантувати цілісність документа (що документ не був змінений після підписання) та підтвердження, що підпис стався від конкретної особи.

4. *Відкликання сертифікатів.* Якщо приватний ключ стає скомпрометованим або з іншої причини сертифікат більше не є дійсним, то він може бути відкликаний через CRL, що дає змогу уникнути зловживань.

Генерація ключів:

Спочатку підписант створює пару ключів: приватний ключ та публічний ключ. Приватний ключ зберігається в безпечному місці (наприклад, на токені або смарт-карті), тоді як публічний ключ може бути наданий для перевірки підписів.

Пара ключів генерується за допомогою криптографічних алгоритмів, таких як RSA або ECDSA (Elliptic Curve Digital Signature Algorithm).

Процес підписання:

1. Підписант створює хеш (унікальне цифрове представлення) документа:

– Для підписання документа спочатку обчислюється його хеш – це цифровий відбиток документа, що дозволяє зменшити його розмір. Зазвичай використовуються хеш-алгоритми, такі як SHA-256 або SHA-3.

– Хеш є унікальним для кожного документа, і будь-які зміни в документі призведуть до зміни хешу.

2. Цей хеш шифрується приватним ключем підписанта, створюючи цифровий підпис:

– Після того, як хеш обчислений, він шифрується за допомогою приватного ключа підписанта. Цей процес створює цифровий підпис, який додається до документа.

– Шифрування хешу гарантує, що підпис може бути перевірений тільки за допомогою відповідного публічного ключа.

Перевірка підписаного документу:

1. Для перевірки підпису отримувач документу використовує публічний ключ підписанта, який можна отримати через сертифікат (наприклад, X.509).

2. Спочатку обчислюється хеш документа, який перевіряється з хешем, що знаходиться в цифровому підписі.

3. Якщо хеші збігаються і підпис можна правильно розшифрувати за допомогою публічного ключа, це підтверджує автентичність підпису і те, що документ не був змінений після підписання.

Хешування має кілька важливих властивостей:

1. *Детермінованість*. Для однакових вхідних даних хеш-функція завжди генерує однакове хеш-значення.

2. *Неможливість відновлення (односторонність)*. З хеш-значення неможливо відновити оригінальні вхідні дані. Це робить хешування корисним для зберігання паролів, оскільки навіть якщо хеш буде вкрадений, не можна безпосередньо дізнатися сам пароль.

3. *Маленька зміна вхідних даних викликає великі зміни в хеші*. Якщо навіть одна літера у вхідних даних зміниться, хеш-значення зміниться кардинально.

4. *Унікальність*. Добре спроектовані хеш-функції повинні мінімізувати ймовірність того, що два різних набори даних дадуть однакові хеші. Такий випадок називається колізією.

Найпоширеніші криптографічні алгоритми [17]:

*RSA (Rivest–Shamir–Adleman).* Один з найпоширеніших алгоритмів для підписів і шифрування, використовується з хеш-функцією (наприклад, SHA-256) для створення цифрових підписів.

*ECDSA (Elliptic Curve Digital Signature Algorithm)*. Використовується в рамках алгоритмів на основі еліптичних кривих. Він надає більшу безпеку при меншому розмірі ключа в порівнянні з RSA.

*SHA-2 (Secure Hash Algorithm 2)*. Використовується для обчислення хешу документів. SHA-256 є однією з найбільш поширених функцій хешування для підписів.

Цифрові підписи використовуються в багатьох сферах:

– Юридичні документи: Контракти, угоди, та інші юридичні документи.

– Фінансові транзакції: Банківські та фінансові звіти, податкові декларації.

– Урядові послуги: Подача заявок на отримання ліцензій або взаємодія з урядовими установами.

– Медицина: Електронні медичні записи та рецепти.

– В Україні для використання цифрових підписів є державні сертифікаційні центри, які надають послуги:

– Видача сертифікатів електронного підпису.

– Перевірка дійсності підпису.

– Використання ключів для криптографічних операцій.

– Сервіси для зберігання і управління ключами.

*Кваліфікований електронний підпис (КЕП)* – це електронний підпис, який має юридичну силу, що прирівнюється до власноручного підпису, відповідно до законодавства. Для того, щоб підпис був кваліфікованим, він повинен відповідати певним вимогам, встановленим національними або міжнародними стандартами (зокрема, Регламентом ЄС № 910/2014, який визначає принципи використання електронних підписів в ЄС). У більшості країн, зокрема в Україні, цей підпис регулюється законодавством, яке забезпечує його юридичну значимість.

Генерація ключів в Україні:

– Для кожного користувача сертифікаційний центр генерує пару криптографічних ключів – публічний і приватний.

– Приватний ключ залишається в безпеці на сервері сертифікаційного центру або в апаратних засобах (токенах, смарт-картах).

– Публічний ключ надається у вигляді сертифіката користувача, який може бути використаний для перевірки підписів.

Використання електронного підпису на веб-сайті.

В Україні, приватний ключ недоступний для передачі на сервер – це правильна і безпечна практика. Приватний ключ не повинен залишати клієнтську сторону, де він зберігається. Це важливо для забезпечення конфіденційності та цілісності електронного підпису. В такому випадку, підписування виконується на клієнтській стороні, а сервер тільки перевіряє підпис та зберігає [14, 15].

Створення документу, який буде збережений на веб сервері разом із підписом та статусом перевірки підпису:

1. Створимо HTML-форму, шаблон документа, яку клієнт буде заповнювати. Кожне поле відповідає певному полю в базі даних для конкретного шаблону документа.

2. Після заповнення форма зберігається у базі даних. Кожен шаблон документа має свою таблицю, а кожен запис в таблиці – це дані з конкретної форми.

3. Клієнт завантажує документ із сервера. Перетворимо документ, який розташований у базі даних, у формат JSON, та повернемо клієнту.

4. Отриманий документ підписується клієнтом. Отриманий підпис клієнт завантажує назад на сервер.

5. Підпис зберігається на сервері в базі даних, і створюється зв'язок між документом і підписом.

6. Сервер перевіряє підписаний документ та встановлює мітку-результат до підпису.

Автоматизація підписання документа.

Щоб спростити алгоритм підписання документа можна використати систему вебхуків. Вебхук – спосіб інтеграції, де один сервіс або система може автоматично сповіщати іншу систему про події (наприклад, про запит на підписання документа), щоб виконати певні дії [14, 15]. Щоб реалізувати таку систему необхідно:

1. Сервер, який містить документ, генерує JSON файл та надсилає запит на сторонній сервіс (наприклад, сервіс для електронного підпису або іншу зовнішню систему), щоб підписати цей документ. В запиті є реквізити отримувача та реквізити сервера, якому необхідно повернути підпис.

2. Клієнт, користуючись стороннім сервісом, підписує документ. Сервіс повертає підпис назад на сервер, який вказаний у реквізитах.

3. Отримавши сповіщення, сервер збереже підпис у базі даних та виконає перевірку підписаного документа.

**Висновки.** У результаті дослідження використання криптографії як сервісу у веб-програмуванні можна зробити висновок, що цей підхід є перспективним інструментом для забезпечення високого рівня безпеки веб-додатків. Завдяки PKI розробники можуть інтегрувати складні криптографічні алгоритми без необхідності глибоких знань у цій галузі, що дозволяє зменшити час на розробку та підвищити надійність систем. Використання хмарних криптографічних сервісів дозволяє оптимізувати процеси обробки даних, зменшити навантаження на сервери та забезпечити більшу гнучкість у масштабуванні веб-додатків. Проте важливо враховувати можливі ризики, такі як безпека зберігання ключів та залежність від постачальника PKI. Для максимізації ефективності необхідно ретельно обирати постачальників послуг, враховуючи їх репутацію, рівень захисту даних і відповідність стандартам безпеки.

**Список використаних джерел:**

1. "Secure Electronic Signature Regulations SOR/2005-30". Justice Laws Website. 10 March 2011. Archived from the original on 28 February 2020. Retrieved 19 May 2020.

2. "US ESIGN Act of 2000" (PDF). Archived (PDF) from the original on 2011-05-22. Retrieved 2006-05-10.

3. Bellare, Mihir; Goldwasser, Shafi (July 2008). "Chapter 10: Digital signatures". Lecture Notes on Cryptography (PDF). p. 168. Archived (PDF) from the original on 2022-04-20. Retrieved 2023-06-11.

4. Ellis, James H. (January 1970). "The Possibility of Secure Non-Secret Digital Encryption" (PDF). Archived from the original (PDF) on 2014-10-
5. Hash_RC6 – Variable length Hash algorithm using RC6 https://ieeexplore.ieee.org/document/7164747 https://wiki.court.gov.ua/w/Що_таке_КЕП_та_ЕЦП%3F
6. JSON – Introduction https://www.w3schools.com/js/js_json_intro.asp
7. Katz Jonathan, Lindell Yehuda. "Chapter 12: Digital Signature Schemes". Introduction to Modern Cryptography. 2007. p. 399.
8. RSA Security's Official Guide to Cryptography by Steve Burnett, Stephen Paine, ISBN-13:978-0072131390, April 19, 2001.
9. Understanding Cryptography: A Textbook for Students and Practitioners by Christof Paar, ISBN-13: 978-3642041006, November 27, 2009.
10. Webhooks https://developer.atlassian.com/server/jira/platform/webhooks/.
11. What is PKI? https://www.digicert.com/what-is-pki.
12. Winn, Jane K. Wright, Benjamin "Digital Signatures: A Survey of Law and Practice in Global Perspective". Journal of Information Technology Law, 2021. Volume 25, Issue 3, pp. 45–60.
13. Головій Л. В., Янчук Ю. В. Правове регулювання інформаційних відносин у сфері електронної комерції. *Право. Людина. Довкілля*. 2020 Том 11, №2. С. 150–157.
14. ЗАКОН УКРАЇНИ Про електронну ідентифікацію та електронні довірчі послуги https://zakon.rada.gov.ua/laws/show/2155-19#Text.
15. Роз'яснення законодавства у сфері ЕДП https://czo.gov.ua/edp-legislation-clarification.
16. Що таке КЕП та ЕЦП?. 17Якими бувають електронні підписи? https://ca.diia.gov.ua/faq17