УДК 004.056.5:004.056.8

АНАЛІЗ СУЧАСНИХ МЕТОДІВ ШИФРУВАННЯ

М. А. Ласкавчук, Л. В. Загоруйко

Анотація. У цій роботі розглянуто сучасні методи шифрування, проведено порівняльний аналіз алгоритмів, внаслідок якого зроблено висновок про найбільш популярний і використовуваний метод шифрування.

Ключові слова: методи шифрування, захист інформації, асиметричне шифрування, симетричне шифрування.

Вступ. Сьогодні людина дедалі більше прив'язана до інформаційного простору, оскільки в ньому суспільство зберігає величезний обсяг персональної та конфіденційної інформації. Політика конкуренції диктує вимогу володіти унікальними даними, які повинні надійно захищатися. Через ці та низку інших причин захисту даних і відомостей приділяється велика увага.

Актуальність. У сучасному світі шифрування є найбільш затребуваним і універсальним засобом захисту конфіденційних даних користувача. Процес шифрування являє собою зміну структурної бази інформаційних даних, розшифрувати яку буде можливо за наявності заздалегідь підготовленого ключа шифрування. Ключ розшифрування відомий лише тому, хто складав шифр, і одержувачу інформації. Шифрування надає захист від різних видів діяльності зловмисників: крадіжка інформації; розкриття засекречених даних; підробка даних під оригінал.

Виклад основного матеріалу. Методи шифрування поділяють на дві категорії [1] – асиметричні та симетричні методи.

Симетрична криптосистема (із закритим ключем, одноключова) – криптосистема, у якій один і той самий алгоритм, а також один і той самий ключ використовується для шифрування та дешифрування повідомлень (рис. 1) [1].

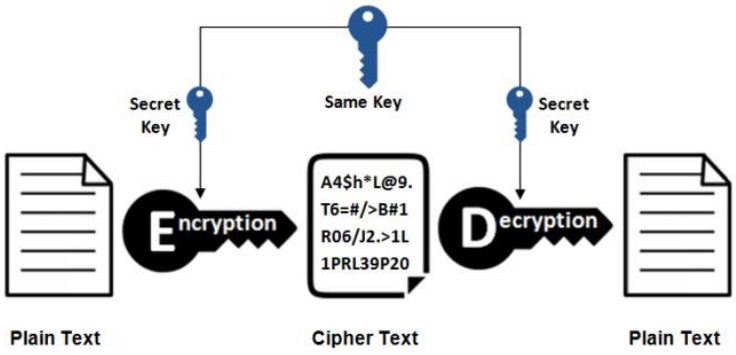

Рис. 1. Схема використання методу симетричного шифрування [1]

Асиметрична криптосистема (з відкритим ключем, двоключова) – криптосистема, у якій використовуються два ключі – відкритий (публічний) і закритий (секретний), які математично пов'язані один з одним. Повідомлення зашифровується з допомогою відкритого ключа, що доступний усім охочим, а розшифровується з допомогою закритого ключа, відомого тільки одержувачу (рис. 2) [1].

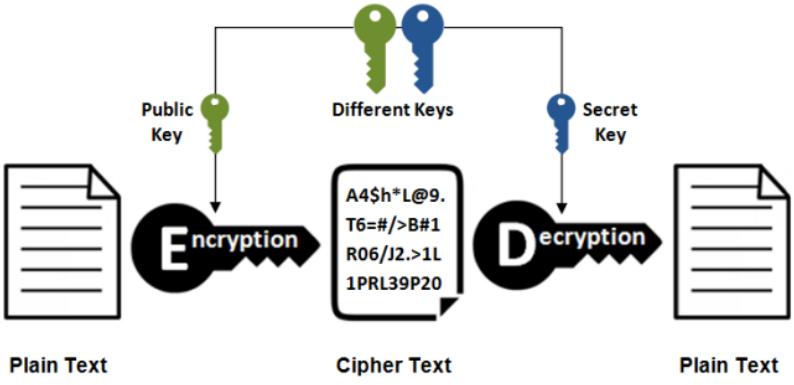

Рис. 2. Схема використання методу асиметричного шифрування [1]

Симетричне шифрування застосовується в сукупності з низкою алгоритмів: *AES, 3DES, CAST, Blowfish, DES.* Асиметричне шифрування використовує такі алгоритми: *El Gamal* і *RSA.*

Методи асиметричного шифрування. *AES* – розроблений у 1997 р., є стандартним методом шифрування на території США. Алгоритм діє на основі симетричного блочного шифру. Реструктуризація даних, яку здійснює алгоритм *AES,* складається з таких завдань [2]:

– *Mix Columns* – перемішування інформації всередині блоку даних;
– *Sub Byties* – підстановка і підбір байтів за допомогою таблиці підстановок;
– *Shift Rows* – зміщення рядків на різні рівні;
– *Expand Key* – виявлення ключів для етапів розшифрування.

Алгоритм *AES* застосовує ключі, довжина яких становить від 128 до 256 біт, і взаємодіє з блоком інформації, довжина якого не перевищує 128 біт [2].

DES – найперший симетричний алгоритм шифрування, розроблений компанією IBM у 1972 р. Шифрування даних на основі цього алгоритму до 2001 р. було стандартом системи шифрування у США. На основі симетричного методу шифрування одержувач і відправник мають лише один ключ для шифрування і розшифрування інформації. Порівняно з попереднім алгоритмом, *DES* працює лише з ключами довжиною в 56 біт, що є його основним мінусом [2].

Стандарт потрійного шифрування даних *(3DES),* відомий як алгоритм потрійного шифрування даних *(TDEA),* вперше запропонований компанією IBM у 1998 р. Кардинальною відмінністю є час, що витрачається на створення шифру, що підвищує ступінь захисту даних [2].

CAST – алгоритм шифрування, що застосовується до блоків даних, які не фіксують. Мають S-блоки зі входом у 8-біт і виходом у 32-біт. Складний в освоєнні, тому не здобув особливої популярності. Працює з ключами 128 і 256 біт [2].

Blowfish – симетричний шифр зі змінною довжиною ключа. Довжина ключа для Blowfish варіюється від 32 біт до 448 біт, він є швидшим за *DES*. Цей алгоритм шифрування найчастіше використовується в разі шифрування великих обсягів інформації, складна система перетворення унеможливлює метод злому блоків із застосуванням перебору ключів. Однак алгоритм стає вразливим для розшифрування в 126 випадках, коли ключ шифрування має високу частоту змін, а робота шифрування будується на невеликих обсягах даних [2].

Методи асиметричного шифрування базуються на двох основних алгоритмах – *El Gamal* і *RSA.*

Алгоритм шифрування *El Gamal* слугує інструментом для створення єдиного ключа шифрування, для шифрування інформації та для проєктування цифрових підписів. Так само може слугувати інструментом ідентифікації [3].

1. Генерування ключів. Генерується просте випадкове число p.

Вибирається генератор g, такий, що $1 < g < p - 1$ та $g^{p-1} mod\ p = 1$ [3].

Вибирається випадкове число x, таке, що $1 < x < p - 1$ [3].

Обчислюється y за формулою [3]:

$$y = g^x mod\ p, \quad (1)$$

де g – генератор;

p – випадкове число;

x – випадкове число.

Відкритими даними є p, g, y.

Закритим ключем є x.

2. Шифрування. Повідомлення M шифрується в такий спосіб [3]:

Вибирається сесійний ключ – випадкове число k, таке, що $1 < k < p - 1$.

Потім обчислюються a та b за формулами [3]:

$$a = g^k mod\ p, \quad (2)$$

де g – генератор;

k – сесійний ключ;

p – випадкове число.

$$b = y^k M\ mod\ p, \quad (3)$$

де y – публічний ключ;
k – сесійний ключ;
M – відкритий текст;
p – випадкове число.
Пара чисел (a, b) є шифротекстом.

3. Дешифрування. Для дешифрування (a, b) обчислюється M за формулами [3]:

$$M = b(a^x)^{-1} \, mod \, p, \tag{4}$$

де b та a – пара чисел, які є шифротекстом;
x – приватний ключ;
p – випадкове число.

$$M = b(a^x)^{-1} \, mod \, p == b \cdot a^{(p-1-x)} mod \, p, \tag{5}$$

де b та a – пара чисел, які є шифротекстом;
x – приватний ключ;
p – випадкове число.

Алгоритм *RSA* є наочним прикладом асиметричного шифрування із застосуванням двох видів ключів – закритого і відкритого. Повідомлення, зашифроване відправником за допомогою цього алгоритму, може прочитати лише одержувач. Це зручно у використанні моделі взаємодії більшої кількості адресатів. Реалізація *RSA* інтенсивно використовує модулярну арифметику, теорему Ейлера та функцію Ейлера. Кожен крок алгоритму включає лише множення, тому його легко виконувати на комп'ютері:

1. Генерування ключів. Спочатку одержувач вибирає два великі прості числа p і q. Їх добуток, що обчислюється за формулою (6), буде половиною відкритого ключа [4].

$$n = p \cdot q, \tag{6}$$

де p і q – просте число.

Одержувач обчислює $\varphi(n)$ за формулою (7) і вибирає число e, відносно просте до $\varphi(n)$. На практиці за e часто вибирають $2^{16} + 1 = 65537$, хоча в деяких випадках воно може бути меншим за 3. Число e буде другою половиною відкритого ключа [4].

$$\varphi(n) = (p - 1)(q - 1), \tag{7}$$

де p і q – просте число.

Одержувач обчислює обернене за модулем число d до e за модулем $\varphi(n)$ за формулою (8) [4]:

$$de = 1(mod \, \varphi(n)), \tag{8}$$

де $\varphi(n)$ – функція Ейлера;
d – закритий ключ.

Одержувач поширює обидві частини відкритого ключа: n і e. Число d тримається в таємниці [4].

2. Шифрування. Для того, щоб зашифрувати текст ***m,*** необхідно обчислити таку рівність за формулою (9) [5]:

$$c \; = \; m^e \cdot (mod \, n), \tag{9}$$

де e – перша частина публічного ключа;
n – друга частина публічного ключа;
m – відкритий текст.

3. Дешифрування. Для того, щоб розшифрувати текст ***c,*** необхідно обчислити таку рівність за формулою (10) [5]:

$$m \; = \; c^d \cdot (mod \, n), \tag{10}$$

де d – приватний ключ;
n – друга частина публічного ключа;
m – відкритий текст.

Висновки. Розглянуто сучасні методи шифрування, які включають у себе симетричні та асиметричні шифри. Симетричне шифрування використовує один ключ для шифрування та

розшифрування даних, тоді як асиметричне шифрування використовує два ключі – приватний та публічний. Один із найпоширеніших алгоритмів симетричного шифрування – *AES*. Він використовує один ключ для шифрування та розшифрування, і його можна налаштовувати на різні довжини ключа (наприклад, 128, 192 або 256 біт).

Наведено принципи роботи алгоритмів асиметричного шифрування *RSA* та *El Gamal,* а саме генерація ключів, процеси шифрування та дешифрування.

Abstract. This article discusses modern encryption methods, conducts a comparative analysis of algorithms, and concludes that the most popular and used encryption method is the most popular.

Keywords: encryption methods, information security, asymmetric encryption, symmetric encryption.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Щур Н. О., Покотило О. А. Основи криптології: навч. посібник. Житомир: Державний університет «Житомирська політехніка», 2021. 120 с. (дата звернення: 15.02.2024).

2. A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security. URL: http://surl.li/qocjd (дата звернення: 15.02.2024).

3. Лабораторна робота № 6. Асиметричні шифри RSAта Ель-Гамаля. Алгоритм обміну ключами Діффі–Хелмана. *Державний університет «Житомирська політехніка» – освітній портал.* URL: https://learn.ztu.edu.ua/pluginfile.php/272225/mod_resource/content/1/%D0%9B%D0%B0%D0%B16.pdf (дата звернення: 16.02.2024).

4. RSA Encryption. URL: https://brilliant.org/wiki/rsa-encryption/ (дата звернення: 16.02.2024).

5. Алгоритм шифрування RSA, види атак на нього. Реалізація мовою Python. URL: https://dou.ua/forums/topic/43026/ (дата звернення: 16.02.2024).

УДК 004.774.6:[004.853:519.2]

МАШИННЕ НАВЧАННЯ ДЛЯ ПЕРСОНАЛІЗАЦІЇ ВЕБКОНТЕНТУ

М. Р. Левченко, І. О. Сеник

Анотація. Сучасний вебконтент вимагає максимальної персоналізації даних, яка вирішується веброзробниками за допомогою штучного інтелекту (ШІ) та машинного навчання (МН). ШІ дає змогу комп'ютерам виконувати завдання, характерні для людського інтелекту, а МН допомагає системам вчитися та покращувати результати без явного програмування. Персоналізація контенту, стимульована ШІ, передбачає адаптацію контенту до індивідуальних інтересів, сприяючи позитивному користувацькому досвіду. Платформи, що базуються на ШІ, включно з генерацією природної мови, дають змогу розробникам масштабувати створення персоналізованого контенту. Персоналізація завдяки передовим технологіям перетворює взаємодію з користувачем із загальної на індивідуалізований досвід. Майбутнє персоналізації вебконтенту залежить від прогресу в розвитку ШІ та МН, що відкриває нові можливості для покращення залученості, лояльності та конверсії користувачів. Використання ШІ та МН вже зараз відкриває широкі перспективи для персоналізованого вебконтенту.

Ключові слова: персоналізація, вебконтент, штучний інтелект, машинне навчання, веброзробники.

Вступ. У епоху сучасного цифрового середовища споживачі все більше очікують, щоб їхні онлайн-враження були максимально персоналізованими та відповідали їх унікальним потребам і вподобанням. Для веброзробників важливо не лише розуміти ці очікування, але й активно використовувати штучний інтелект та технології машинного навчання для досягнення цієї мети.

Штучний інтелект – це напрям розвитку комп'ютерних систем, які здатні виконувати завдання, що традиційно вимагають людського інтелекту. Серед таких завдань можуть бути візуальне сприйняття, розпізнавання мови, прийняття рішень та мовний переклад [1].

Машинне навчання – це розділ штучного інтелекту, який дає змогу комп'ютерним системам навчатися на даних і покращувати свої результати без явного програмування (не потрібно писати інструкції для кожної задачі). Воно працює на основі алгоритмів (спеціальних правил), які аналізують великі обсяги інформації, шукають у ній закономірності та використовують ці знання для ухвалення рішень або прогнозування [1].

Основний розділ. Персоналізація контенту являє собою стратегію використання зібраної інформації про аудиторію для індивідуалізації пропонованого контенту відповідно до її унікальних інтересів та запитів. Цей підхід дає змогу створювати більш особистий та затишний