

АЛГОРИТМ КРИПТОГРАФІЧНОГО ПРОТОКОЛУ АВТЕНТИФІКАЦІЇ БЕЗ ПЕРЕСИЛАННЯ ПАРОЛЮ МЕРЕЖЕЮ У ЯВНОМУ ВИГЛЯДІ

Стрельніков М. Д.¹; Партас В. К., к.т.н²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Україна, Миколаїв

¹211227@nuos.edu.ua; ²viktor.partas@nuos.edu.ua

Анотація. У роботі приділено увагу до основних проблем мережевої безпеки при пересиланні паролів у поширених алгоритмах автентифікації. Зокрема проаналізована вразливість відкритої пересилки пароля до пасивних мережевих атак, атак типу “людина посередині” та загроза компрометації паролю, якщо він зберігається на сервері у відкритому вигляді. Розглянуто криптографічний алгоритм SPAKE2, що уникає пересилки пароля у відкритому вигляді та генерації разових сесійних спільних ключів.

Ключові слова: автентифікація; PAKE; повідомлення; пароль; криптографія.

Вступна частина. Актуальність даної роботи полягає у дослідженні способів, зменшення ризиків мережевої безпеки у протоколах автентифікації, які передбачають пересилку паролю відкритими каналами зв'язку. Патерн автентифікації із відправленням паролю публічною мережею на сервер є типовим для більшості сучасних WEB-додатків та сервісів. Велика кількість таких сервісів є дуже критичною — це можуть бути банки, ПО серверів бухгалтерії, особисті електронні кабінети, навіть соціальні мережі тощо. Подібні сценарії становлять великий рівень ризику компрометації чи втрати секретності пароля, як при активних, так і за пасивних мережевих атаках. У роботі зроблено огляд та аналіз розповсюджених векторів атаки на пароліні протоколи автентифікації, зокрема “людина посередині”. Запропоновано використовувати криптографічні протоколи, що дозволяють уникати передавання паролів у відкритому вигляді мережею.

Основна частина. Автентифікація — процедура встановлення належності користувачеві інформації в системі на основі пред'явленого ним ідентифікатора («логіна» — реєстраційного імені користувача) і пароля — певної конфіденційної інформації, знання якої передбачає володіння певним ресурсом в мережі. Протоколи авторизації за паролем реалізовані наступним чином у більшості клієнт-серверних додатків:

1. Клієнт відправляє на бік сервера логін та пароль;
2. Сервер отримує логін та пароль користувача;
3. Сервер зіставляє за базою даних логін із паролем, якщо пароль є валідний для логіна певного користувача — користувач проходить авторизацію.

Розглянемо наведену реалізацію алгоритму авторизації з точки зору мережевої безпеки. Можливі вектори атаки:

- Мережа та стек мережевих протоколів, що реалізують пересилку чутливих даних під час виконання алгоритму авторизації;
- Сервер бази даних, що зберігає ідентифікатори користувачів та резервні копії даних, СУБД;
- Серверна файлова та операційна системи, критичне системне ПО, засоби віддаленого адміністрування сервера.

В залежності від реалізації бази даних, неавторизований користувач може виконати SQL-ін'єкцію[1] (чи експлуатувати іншу вразливість СУБД) для викрадення авторизаційних даних певного користувача (чи вмісту всієї бази у гіршому випадку) та отримати неавторизований доступ до комп'ютерної системи. Потенційний зловмисник може використати вразливості конфігурації засобів віддаленого адміністрування (напр. SSH, SSH+LDAP, telnet) чи іншого критичного ПО серверу, щоб отримати доступ до файлової системи та резервних копій бази. Можна побачити, що зі складністю системи дуже швидко зростає кількість вразливих місць, а для реалізації вектору атаки буде достатньо експлуатації лише однієї вразливості. Проте, можливо набагато зменшити площу атаки на системні компоненти шляхами:

- Налаштування детального моніторингу системи, логінгу подій та проведення регулярних аудитів комп'ютерної системи та її компонентів;
- Безпечне конфігурування засобів віддаленого адміністрування, своєчасне встановлення оновлень ОС та драйверів та закриття мережевого периметру комп'ютерної системи у приватній мережі чи за VPN;
- Налаштування параметрів безпеки СУБД та БД (логінг усіх запитів, обмеження максимальної кількості під'єднань, їх тривалості, встановлення ролей користувачів БД мінімально достатніх для виконання саме їх задач та ін.);
- Шифрування файлової системи та резервних копій.

Перелік цих мір значно підвищує рівень безпеки БД у комп'ютерних системах, та підвищує загальну надійність ресурсів системи, але ускладнює комп'ютерну систему, знижує загальну швидкодію та може впливати на зручність використання. На практиці намагаються завжди відповідати компромісам, забезпечуючи і гідний рівень безпеки, і зручність. По при все, залишається фундаментально слабе місце у системі — це те, що усі авторизаційні дані зберігаються в одному місці і всі компоненти системи від нього залежать. Одночасно із цим, комп'ютерна система залишається вразливою для внутрішніх атак. Вони можуть бути, як цілеспрямованими, так і випадковими: через людський фактор та необережність, інфікування комп'ютерним вірусом обладнання, фішинг спрямований на користувачів системи чи засоби соціальної інженерії та ін. Є досить ефективне рішення: замість паролів, у базі даних зберігаються геш-значення паролів. У цьому випадку, навіть отримання неавторизованого доступу до вмісту всієї БД не призведе до катастрофічних наслідків. Але для цього потрібно слідкувати наступним рекомендаціям:

- Для формування геш-значень з паролів повинні бути використані криптографічні геш-функції, які щонайменше відповідають стандарту SHA-2 (лише SHA-256 та SHA-512, вважаються відносно надійними на сьогодні), чи краще — SHA-3 (Кессак та SHAKE);
- Для додаткового підсилення криптографічної надійності, гешування може бути виконано із додаванням “солі” — короткого рядка даних, який додається до даних на вході геш-функцій, щоб навіть за умови підбору колізії геш-функції неможливо було відновити пароль.

Але залишається загроза використання зловмисником мережевого вектора атаки (пасивної чи активної). Сценарій пасивної мережевої атаки передбачає, що конфіденційний трафік передається відкритою публічною чи домашньою мережею користувача, а потенційний зловмисник має доступ до цієї мережі. Наприклад, він може фізично під'єднатись до мережі потенційної жертви через Wi-Fi або навіть до Ethernet-кабелю за допомогою спеціального обладнання. Зловмиснику достатньо лише використати ПО, що

виконує сканування TCP-пакетів мережі (напр. Wireshark) і він отримає дані про логіни і паролі потенційних жертв у відкритому вигляді. Такі сценарії мають місце у домашніх мережах, у загальнодоступних публічних мережах чи усередині мережевого периметру певної установи. У останньому випадку, адміністратори мереж дуже рідко вмикають шифрування трафіку усередині корпоративної мережі чи багатофакторну автентифікацію, оскільки припускають, що мережа установи є цілком приватною, а доступ до інтернету відбувається лише через спеціальні шлюзи. Але це не виключає можливості того, що робочі станції користувачів корпоративної мережі будуть заражені вірусами, чи користувачі самі будуть змотивовані провести подібного роду атаку задля власної матеріальної вигоди.

Проблему пасивних мережевих атак можна досить успішно вирішити із увімкненням шифрування трафіку на різних рівнях моделі OSI[2] із використанням певного мережевого протоколу:

- WEP/WPA2/WPA3 (канальний рівень) — протоколи, які реалізують шифрування трафіку на каналному рівні у Wi-Fi мережах;
- Ipsec (мережевий рівень) дозволяє шифрувати трафік будь-якого протоколу, заснованого на IP — його часто використовують для побудови приватних мереж та VPN;
- Kerberos (протокол прикладного рівня UDP у якості транспортного протоколу) дозволяє забезпечити автентифікацію для клієнтів та серверу на рівні сокетів, а також симетричне шифрування каналу зв'язку між комп'ютерами у відкритій мережі;
- HTTPS (протокол прикладного рівня для безпечного доступу до WEB-ресурсів через браузер із TCP у якості транспортного протоколу) — розширена версія HTTP протоколу, яка додає функціонал шифрування вмісту HTTP-пакетів за допомогою сесійного ключа шифрування, узгодженого на етапі автентифікації.

Використання наведених протоколів дозволяють значно підвищити рівень безпеки мереж та майже унеможливити пасивні мережеві атаки. Проте залишається загроза активних мережевих атак, таких як “людина посередині” (англ. Man-in-the-middle)[3] та атак на мережеве обладнання. Слід зауважити, що протоколи WEP/WPA2 використовують застарілі на сьогодні стандарти шифрування (RC4), а також вразливість паролів Wi-Fi до атак прямим перебором чи атакою за словником. WPA3 на сьогодні є більш довершеним протоколом з точки зору безпеки, що протидіє також атакам прямим перебором, але ще недостатньо поширеним серед виробників мережевого обладнання. Протоколи шифрування трафіку інших рівнів моделі OSI теж мають свої вразливості. Наприклад, конфігурація протоколу Kerberos на комп'ютерах під керуванням ОС Windows за замовченням містить режим автентифікації із використанням NTLM-гешування[4], але NTLM-геші передаються відкритою мережею і є вразливим до атак із використанням “райдужних таблиць” (метод пошуку колізій геш-функцій, що використовує попередньо згенеровану множину довільних геш-функцій). Дуже поширеним є використання HTTPS — його використовують більшість WEB-додатків. Через поширеність протоколу, в ньому забезпечено зворотню взаємодію із молодшими версіями, але це є слабкою стороною протоколу. Через це існує велика кількість атак, що зосереджується саме на пониженні версії протоколу до мінімальної та подальшу експлуатацію відомих вразливостей (наприклад встановлення слабких криптографічних параметрів). Безпечнішою вважається остання версія протоколу HTTP, що включає у себе TLS1.3[5]. Безумовно, використання засобів безпеки на рівні мережевих протоколів ще значно зменшує площу потенційної атаки, але все одно не виключає такої можливості. Згідно MITRE ATT&CK (глобальна та публічна база знань задокументованих

вразливостей)[6], в HTTPS по усім існуючим версіям задокументовано близько 700 вразливостей.

Для зниження ризику розкриття паролю під час його передачі було модифіковано використання паролів у протоколах авторизації та сформульовано протоколи типу PAKE (англ. Password-Authenticated Key Exchange)[7]. Сутність PAKE полягає у тому, що сторонам заздалегідь відомо геш-значення паролю одного з учасників, тоді сторони можуть незалежно випрацювати спільний сесійний ключ. У такому випадку авторизація буде вважатися успішною, коли ключ, випрацьований сторонами за алгоритмом співпадає. Геш-значення оригінального паролю може бути сформовано при реєстрації чи за допомогою протоколу асиметричного розподілу ключів (напр. Діфі-Гелмана[8]). Серед протоколів PAKE є різні реалізації (EKE, J-PAKE, SPAKE1/2, OPAQUE та ін.). У роботі розглянуто реалізацію PAKE за стандартом SPAKE2[9], який працює на еліптичних кривих, як і більшість сучасних засобів мережевої криптографії із відкритим ключем, а саме на кривих Едвардса[10] (за стандартом Ed25519)[11]. Криві Едвардса забезпечують простішу реалізацію дій складання точок із одночасним забезпеченням високого рівня криптографічної стійкості. Крива Едвардса (рисунок 1.1) задається рівнянням $x^2 + y^2 = 1 + dx^2y^2$ (параметр $d \neq 0$ — інакше крива Едвардса стане колом).

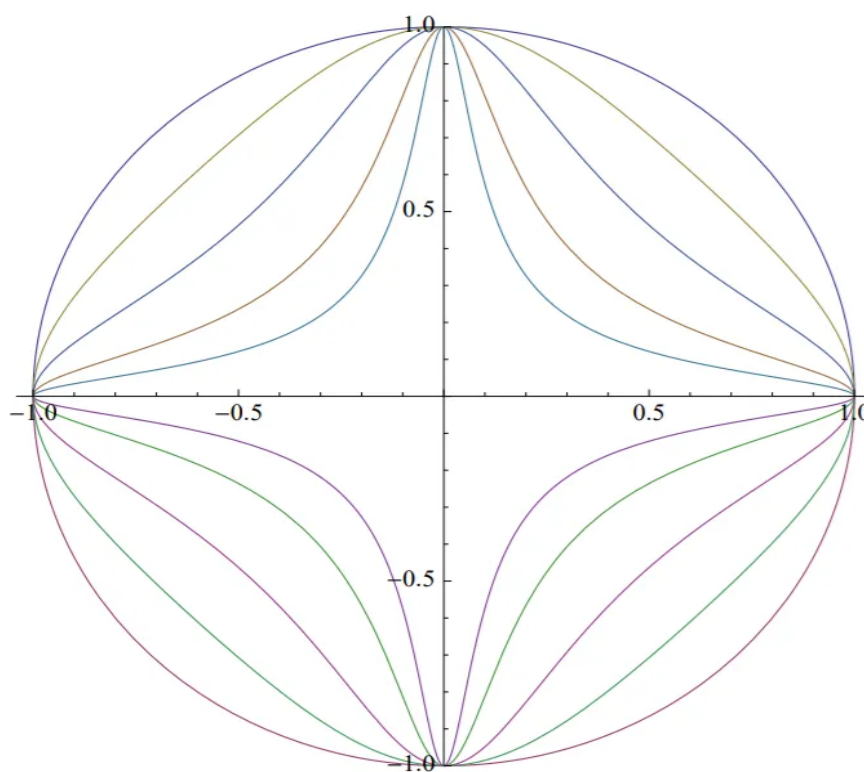


Рисунок 1.1 Сімейство кривих Едвардса .

Складання на кривих Едвардса визначається формулою (1).

$$(x_A, y_A) + (x_B, y_B) = \left(\frac{x_A y_B + x_B y_A}{1 + dx_A x_B y_A y_B}, \frac{y_A y_B - x_B x_A}{1 - dx_A x_B y_A y_B} \right). \quad (1)$$

У свою чергу, множення точки на число виконується, як багатократне складання, викрнюючі ітерації згідно формулі (1).

Параметри на вході алгоритму (відомі усім учасниками та не є секретними): публічні ключі A та B — довільні точки M та N спеціальної кривої Едвардса Ed25519; велике просте число p , G — базова точка кривої (обирається довільно)

Усі розрахунки проводяться за модулем p , а складання та множення визначені відповідно до формули (1).

Припустимо, що в нас є два абонента A та B . Тоді алгоритм виглядає наступним чином. Дії абонента A :

1. A знає спільний пароль $pass$, переводячи його до числа w за допомогою геш-функції;
2. A обирає випадковим чином число x у проміжку $[0; p]$;
3. A розраховує точки X та T згідно формули (1);
4. A відправляє точку T абоненту B .

$$\begin{aligned} w &= \text{hash}(pass) \\ x &= \text{rand}(0, p) \\ X &= xG \pmod{p} \\ T &= wM + X \pmod{p} \\ \text{Alice: } T &\rightarrow B \end{aligned} \quad (2)$$

Відповідно B діє наступним чином:

1. B отримує T ;
2. B знає спільний пароль $pass$, переводячи його до числа w за допомогою геш-функції;
3. B також обирає випадковим чином число y у проміжку $[0; p]$;
4. B розраховує Y та S згідно формули (2);
5. B відправляє Y абоненту A .

$$\begin{aligned} \text{Bob: } B &\leftarrow T \\ w &= \text{hash}(pass) \\ y &= \text{rand}(0, p) \\ Y &= yG \pmod{p} \\ S &= wN + Y \pmod{p} \\ \text{Bob: } Y &\rightarrow A \end{aligned} \quad (3)$$

Тепер A та B виконують етап верифікації незалежно один до одного:

1. A розраховує $K_B: K_B = x(S - wN) = x(wN + Y - wN) = xY = xyG \pmod{p}$;
2. B розраховує $K_A: K_A = y(T - wM) = y(wM + X - wN) = yX = xyG \pmod{p}$.

Таким чином, між сторонами A та B узгоджено ключ $K_A = K_B$, який є похідним від паролю, як наслідок пароль не передається мережею. Мережею передаються лише параметри T та Y , які не надають достатньої інформації для відновлення паролю, оскільки для злоумисника залишаються невідомі x та y , що у свою чергу є секретними випадковими числами. А сам пароль $pass$ не приймає участі у розрахунках — замість нього використовується його криптографічний геш w . Тобто навіть користувачеві не потрібно зберігати оригінальний пароль $pass$ на пристрої для авторизації. За подібним алгоритмом також можна генерувати разові сесійні ключі, підсиливши рівень безпеки, шляхом зміни випадкових параметрів x та y у кожній сесії.

Висновки. Використання SPAKE2 у якості криптографічного протоколу автентифікації допомагає уникнути необхідності пересилання та зберігання паролів між клієнтом та сервером у відкритому вигляді. Корисною властивістю протоколу є те, що довжина паролю не має суттєвого значення, оскільки пароль не використовується у чистому вигляді, а тільки його відбиток. Будова протоколу SPAKE2 робить його особливо зручним у формуванні

разових сесійних ключів — це стає можливим із використанням випадкових чисел у алгоритмі генерування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Rajbanshi, S. (2020, July 27). 3 ways of stealing data using SQL injection. <https://medium.com/devil-is-in-the-details/3-ways-of-stealing-data-using-sql-injection-71893672efcf>
- [2] Oracle. (n.d.). Open Systems Interconnection (OSI) Reference Model. <https://docs.oracle.com/cd/E19504-01/802-5886/intro-45828/index.html>
- [3] Cybercalm. (2020, May 26). Що таке атака "Man-in-the-Middle" та як себе захистити? <https://docs.oracle.com/cd/E19504-01/802-5886/intro-45828/index.html>
- [4] Microsoft MSRC. (2022, January 11). Windows Kerberos Elevation of Privilege Vulnerability (CVE-2022-21920). <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-21920>
- [5] Sullivan, N. (2018, November 8). A Detailed Look at RFC 8446 (a.k.a. TLS 1.3). <https://blog.cloudflare.com/rfc-8446-aka-tls-1-3/>
- [6] MITRE ATT&CK. (н.д.). Globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. <https://attack.mitre.org/>
- [7] Chang, C.-M., (н.д.). J-PAKE over TLS. <https://chunminchang.gitbooks.io/j-pake-over-tls/content/>
- [8] Вікіпедія. (н.д.). Протокол Діффі — Геллмана. https://uk.wikipedia.org/wiki/Протокол_Діффі_—_Геллмана
- [9] Buchanan, W. J. (2022, November 4). SPAKE2. <https://asecuritysite.com/encryption/spake>
- [10] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., Yang, B.-Y. (2017, January 22). High-speed high-security signatures. <https://ed25519.cr.yp.to/ed25519-20110926.pdf>
- [11] Housni, Y., Edwards curves. (2018, December 3). hal-01942759. <https://hal.archives-ouvertes.fr/hal-01942759/document>

Algorithm of cryptographic authentication protocol without transmitting the explicit password over network

Abstract. *The work focuses on the main problems of network security when forwarding passwords in common authentication algorithms. In particular, the vulnerability of open password transmission to passive network attacks, man-in-the-middle attacks and the threat of password compromise if it is stored on the server in open form is analyzed. The cryptographic algorithm SPAKE2, which solves the problem of sending a password in the open and generating one-time session shared keys, is considered.*

Keywords: authentication; PAKE; message; password; cryptography.