Galašifrēšana, izskaidrota pa īstam

Ko pakalpojumu sniedzēji saka, kad viņi saka E2EE, un ko viņi noklusē. Didaktisks mehānisma un tā robežu skaidrojums bez reklāmas iepakojuma.

Būsim skaidri: WhatsApp saka, ka jūsu ziņojumi ir galašifrēti. Tā ir taisnība — un ar to nepietiek. Ja dublējums tiek nosūtīts uz iCloud vai Google Drive bez papildu šifrēšanas, šifrēšana tiek salauzta jūsu pašu tālrunī. Operatīvais jautājums nav par to, vai tas ir šifrēts, bet kur atrodas atslēgas.

Ko šifrēšana nozīmē patiesībā

Ziņojuma šifrēšana nozīmē tā pārvēršanu par kaut ko tādu, kas izskatās pēc trokšņa jebkuram, kuram nav noteiktas informācijas, ko sauc par atslēgu. Operācija tiek veikta sūtītāja ierīcē un ar pareizo atslēgu tā tiek atcelta saņēmēja ierīcē. Pa vidu ziņojums ceļo kā baitu secība bez acīmredzamas nozīmes. Tā ir vienkāršā ideja. Pārējā raksta daļa pievēršas niansēm, kas to atkarībā no gadījuma pārvērš par reālu garantiju vai tirgus etiķeti.

Īpašības vārds *galašifrēšana* — angļu valodā *end-to-end*, saīsināti E2EE — piešķir precizitāti. Šifrēšana netiek veikta tāpēc, lai starpnieka serveris varētu to izlasīt un piegādāt. Tā tiek veikta tāpēc, lai tikai abiem galiem — sūtītāja ierīcei un saņēmēja ierīcei — piederētu atslēga. Jebkurš serveris, caur kuru iet ziņojums, redz troksni, nevis ziņojumu. Tā ir tehniskā atšķirība no šifrēšanas *tranzītā*, kur saturs ceļo šifrēts no viena servera uz nākamo, bet katrs serveris, caur kuru tas iet, to atšifrē, lai to pārsūtītu, uz laiku atgūstot atklātu tekstu.

Kopīgā noslēpuma paradokss

Ir acīmredzama problēma. Lai divi cilvēki varētu savā starpā šifrēt un atšifrēt ziņojumus, abiem ir nepieciešama viena un tā pati atslēga. Bet kā viņi vienojas par šo atslēgu, ja viss, ko viņi sūta viens otram, pēc definīcijas iet caur kanālu, kur kāds varētu klausīties? Vienošanās par atslēgu tajā pašā kanālā, kurā viņi to vēlāk izmantos, šķiet neiespējama: ja uzbrucējs to dzirdēs vienošanās brīdī, viņš varēs atšifrēt visu turpmāko. Desmitgadēm ilgi klasiskā kriptogrāfija to risināja smagā veidā: atslēgas tika nodotas personīgi pirms to izmantošanas uzsākšanas fiziskās tikšanās reizēs. Vēstnieki nēsāja koferus ar atslēgām, kas bija iešūtas mēteļa oderē.

Mūsdienu e-pastā šis risinājums nav mērogojams. Ja mums būtu fiziski jādodas uz katra cilvēka mājām, ar kuru mēs plānojam sazināties šifrētā veidā, mēs ne ar vienu nevarētu parunāt. Jautājums, ko kriptogrāfijas kopiena uzdeva pirms piecdesmit gadiem, bija šāds: vai ir iespējams, ka divi cilvēki, kuri nepazīst viens otru un kuriem ir tikai publisks kanāls, tajā pašā publiskajā kanālā vienotos par noslēpumu, ko neviens, kas klausās kanālu, nevarētu uzzināt?

Diffie-Hellman elegance

1976. gadā divi matemātiķi, vārdā Whitfield Diffie un Martin Hellman, pierādīja kaut ko šķietami neiespējamu: ka divi cilvēki, runājot tikai caur publisku kanālu — kanālu, kurā ikviens var dzirdēt visu, ko viņi saka —, var vienoties par slepenu paroli bez jebkāda klausītāja iespējas to atklāt. Tas izklausās pēc maģijas. Tā nav: tā ir matemātika. Diffie-Hellman atslēgu apmaiņa, kā tā kopš tā laika ir zināma, ir pamatā praktiski visai šifrētajai komunikācijai internetā, un pusgadsimtu ilga intensīva lietošana un pasaules akadēmiskā pārbaude apstiprina tās pamatīgumu. Kas vēlas redzēt vizuālo intuīciju vai matemātiku, var turpināt lasīt. Kas dod priekšroku uzticēties, ka tas darbojas, var arī turpināt, nezaudējot raksta pavedienu.

Tiem, kas vēlas to iedomāties attēlā, ir zināma analoģija ar krāsām. Iedomājieties, ka Alise un Bruno publiski vienojas par pamatkrāsu — teiksim, dzeltenu — Evas priekšā, kura viņos klausās. Katrs privāti izvēlas otru slepeno krāsu un sajauc savu noslēpumu ar dzelteno. Alise iegūst noteiktu oranžu; Bruno iegūst noteiktu zaļu. Viņi apmainās ar rezultātiem Evas priekšā. Tagad katrs sajauc saņemto krāsu ar savu noslēpumu, un abi nonāk pie tās pašas galīgās krāsas, jo sajaukšanas secībai nav nozīmes. Eva redzēja dzelteno un abus starpmaisījumus, bet ne noslēpumus; bez kāda no noslēpumiem viņa nevar nonākt pie galīgās krāsas. Reālā matemātika krāsas aizstāj ar kāpināšanu modulārajās grupās vai eliptiskajās līknēs, bet ideja ir tā pati: kopīgais noslēpums tiek izveidots publiski bez neviena iespējas kanālā to rekonstruēt.

Aritmētikā tiem, kas dod priekšroku redzēt mehānismu: Alise izvēlas slepenu skaitli a, Bruno izvēlas b. Viņi apmainās ar g^a un g^b atklāti kanālā. Alise aprēķina $(g^b)^a$, bet Bruno aprēķina $(g^a)^b$; abi nonāk pie viena un tā paša g^{ab}. Eva redz g, g^a un g^b ejot caur kanālu, bet atgūt a no g^a — tā saucamā diskrētā logaritma problēma — prasa astronomisku aprēķinu laiku, kas pārsniedz visuma vecumu, kad g tiek izvēlēts piemērotā matemātiskā grupā.

Para quien quiera comprobarlo con números pequeños. El intercambio Diffie-Hellman se puede recorrer entero con cifras lo bastante reducidas como para hacer las cuentas a mano. Quien prefiera no entrar en aritmética puede saltarse este bloque sin perder el hilo del artículo; quien quiera ver el mecanismo funcionando paso a paso lo encontrará aquí. **Las reglas públicas**, que cualquiera puede leer: un primo $p = 11$ (en el Diffie-Hellman real es de unas trescientas cifras; usamos once para que las cuentas quepan en una página), una base $g = 2$, y la convención de

que toda la aritmética se hace *módulo p* — se calcula, se divide entre *p*, y se conserva el resto, como un reloj de once posiciones que vuelve al cero al rebasar el diez. **Las elecciones privadas**, una cada uno y jamás compartidas: Alicia elige $a = 4$. Bruno elige $b = 7$.

Paso 1. Alicia calcula $2^4 = 16$, luego $16 \bmod 11 = \mathbf{5}$. Envía el cinco. Eva lo anota.

Paso 2. Bruno calcula $2^7 = 128$, luego $128 \bmod 11 = \mathbf{7}$. Envía el siete. Eva también lo anota. Tras los dos envíos, la libreta de Eva contiene cuatro datos: $p = 11$, $g = 2$, A = 5, B = 7. Le falta el número compartido que Alicia y Bruno están a punto de derivar — y que Eva no podrá reconstruir.

Paso 3. Alicia toma el siete que Bruno le envió y lo eleva a su exponente privado $a = 4$. Para evitar manejar $7^4 = 2401$, se calcula por partes aplicando el módulo en cada paso:

$7^2 = 49$

$49 \bmod 11 = 5$

$7^4 = (7^2)^2 = 5^2 = 25$

$25 \bmod 11 = 3$

Alicia obtiene el número **3**.

Paso 4. Bruno toma el cinco que Alicia le envió y lo eleva a su exponente privado $b = 7$. De nuevo por partes:

$5^2 = 25 \bmod 11 = 3$

$5^4 = (5^2)^2 = 3^2 = 9 \bmod 11 = 9$

$5^6 = 5^4 \times 5^2 = 9 \times 3 = 27 \bmod 11 = 5$

Finalmente $5^7 = 5^6 \times 5 = 5 \times 5 = 25 \bmod 11 = \mathbf{3}$.

Bruno obtiene también **3**.

Los dos han llegado al mismo número, 3, trabajando en paralelo. Ninguno envió su exponente privado en ningún momento. Alicia no sabe que $b = 7$; Bruno no sabe que $a = 4$. Cada cual usó el valor público que el otro envió combinado con su propio exponente privado, y se encontraron en el mismo destino. **¿Por qué llegan al mismo número?** Lo que calculó cada uno: Alicia, $(g^b)^a = 2^{7\times4} = 2^{28} \bmod 11$. Bruno, $(g^a)^b = 2^{4\times7} = 2^{28} \bmod 11$. Es la misma cantidad porque el orden de multiplicación de exponentes no importa ($7 \times 4 = 4 \times 7$). Cada cual llegó por un camino distinto al mismo destino.

¿Y Eva? Tiene en su libreta $p = 11$, $g = 2$, A = 5, B = 7, y quisiera el 3. Para calcularlo necesitaría conocer a o b — pero ninguno ha viajado por el canal. Su única vía es preguntarse: «¿para qué exponente a se cumple $2^a \bmod 11 = 5$?». Con p tan pequeño puede probar 0, 1, 2, 3, 4… y encontrarlo en menos de un minuto. Pero al sustituir 11 por un primo de trescientas cifras, el espacio de exponentes posibles tiene más elementos que átomos hay en el universo observable. **No existe a día de hoy ningún algoritmo conocido por la humanidad que pueda recorrer ese espacio en menos de miles de millones de años.** Es el llamado *problema del logaritmo discreto*: fácil hacia adelante, computacionalmente imposible hacia atrás. Y es la razón por la que el cifrado resiste aunque Eva haya seguido toda la conversación letra por letra.

Tres ingredientes simples —aritmética sobre un reloj, exponenciación, y conmutatividad de la multiplicación ($a \cdot b = b \cdot a$)— combinados producen un protocolo del que media humanidad depende cada día para sus comunicaciones privadas. Ninguna de las tres piezas, por separado, parece especial. Lo decisivo es el ensamblaje.

No Diffie-Hellman līdz Signal protokolam

Galašifrēšana, ko šodien izmanto profesionālas ziņapmaiņas lietotnes, gandrīz bez izņēmuma balstās uz elegantu un nostiprinātu Diffie-Hellman apmaiņas versiju. Signal protokols, ko 2013.–2016. gadā izstrādāja Trevor Perrin un Moxie Marlinspike, ir etalons. Tas apvieno divas galvenās idejas. Pirmā — atslēgu apmaiņa eliptiskajās līknēs (X25519), kas izveido sākotnējo kopīgo noslēpumu starp divām ierīcēm. Otrā — tā sauktais Double Ratchet — dubultais sprūdrats —, kas automātiski atjauno atslēgas ar katru ziņojumu, tā ka ierīces kompromitēšana šodien neļauj atšifrēt pagātnes ziņojumus, ne arī nākotnes ziņojumus, kad sprūdrats ir pagriezts.

Zig valodā X25519 apmaiņa, kas izveido kopīgo noslēpumu starp divām ierīcēm, ietilpst sešās rindās, izmantojot standarta bibliotēku:

```
const std = @import("std");
const X25519 = std.crypto.dh.X25519;

// Alicia y Bruno generan cada uno un par (privada, pública).
const par_alicia = X25519.KeyPair.generate(io);
const par_bruno  = X25519.KeyPair.generate(io);

// Cada parte recibe la clave pública de la otra y deriva el mismo secreto.
const secreto_alicia = X25519.scalarmult(par_alicia.secret_key, par_bruno.public_key) catch unreachable;
const secreto_bruno  = X25519.scalarmult(par_bruno.secret_key,  par_alicia.public_key) catch unreachable;
// secreto_alicia == secreto_bruno  (32 bytes)
```

Kas notiek tajās sešās rindās: Publiskās atslēgas ceļo atklāti. Privātās atslēgas nekad nepamet attiecīgo ierīci. Katra puse no savas privātās un otras puses publiskās atslēgas atvasina vienu un to pašu trīsdesmit divu baitu noslēpumu, ko neviens kanālā nevar atgūt. Šis noslēpums vēlāk kalpo kā sākumpunkts apmainīto ziņojumu šifrēšanai. Signal protokola Double Ratchet pievieno pastāvīgu šī materiāla rotāciju, lai viena mirkļa kompromitēšana neapdraudētu pārējo sarunu.

Un kas tieši atrodas `std.crypto.dh.X25519` iekšpusē? Nekādas slēptas maģijas. Tās ir divas īsas funkcijas, kuras pilnībā var izlasīt pašā Zig standarta bibliotēkā. Pirmā no tām atvasina publisko atslēgu no privātās — apmaiņas «g^a»:

```
pub fn recoverPublicKey(secret_key: [secret_length]u8) IdentityElementError![public_length]u8 {
    const q = try Curve.basePoint.clampedMul(secret_key);
    return q.toBytes();
}
```

Raksta valodā: privātā atslēga tiek «reizināta» — eliptiskā, nevis elementārā aritmētiskā nozīmē — ar Curve25519 līknes bāzes punktu, un rezultāts tiek serializēts trīsdesmit divos baitos. Operācija `clampedMul` ir šīs skalārās reizināšanas pastiprinātā versija: tajā ir iekļauti drošības pasākumi, ko kriptogrāfijas kopiena gadu gaitā pievienoja, lai pretotos zināmām uzbrukumu saimēm. Divas funkcijas ķermeņa rindas.

Otrā funkcija apvieno jūsu privāto atslēgu ar publisko atslēgu, ko jums nosūta otra puse. Tas ir apmaiņas «$(g^b)^a$», kas rada trīsdesmit divu baitu kopīgo noslēpumu, ko neviens no jums nekad nav pārraidījis:

```
pub fn scalarmult(secret_key: [secret_length]u8, public_key: [public_length]u8) IdentityElementError![shared_length]u8 {
    const q = try Curve.fromBytes(public_key).clampedMul(secret_key);
    return q.toBytes();
}
```

Vēl divas rindas. Saņemtā publiskā atslēga tiek interpretēta kā punkts uz līknes, un tiek «reizināta» ar paša privāto atslēgu. Pateicoties līknes operācijas komutatīvajai īpašībai — analogi eksponentu reizināšanas komutatīvajai īpašībai, ko redzējām skaitliskajā piemērā —, abas puses iegūst vienu un to pašu serializēto punktu: tieši to kopīgo noslēpumu, par kuru runā raksts.

Tas arī viss. Tas, kas lietotnē izskatās pēc maģijas, patiesībā ir divas funkcijas, katra trīs rindu garumā. Tehniskā sarežģītība ir koncentrēta vienā operācijā, `clampedMul`, kas ir ierakstīta tālāk tajā pašā standarta bibliotēkā, kuru gadu desmitiem ir pārskatījusi starptautiskā kriptogrāfijas kopiena, un kura ir pieejama ikvienam, kurš vēlas to izlasīt burtu pa burtam. Nedz mūsu lietotnē, nedz Zig standarta bibliotēkā nav „melnās kastes". Tur ir atvērtā koda kods, ko cilvēks var saprast, izvēloties ātrumu, kādā vēlas tajā iedziļināties.

Ko aizsargā galašifrēšana

Tas, ko E2EE labi aizsargā, pieņemot pareizu īstenošanu, ir ziņojuma saturs tranzītā. Starpnieka serveris, kas saņem un pārsūta šifrētos datus, redzēs nesaprotamu baitu secību. Uzbrucējs ar piekļuvi kabelim, maršrutētājam vai wifi piekļuves punktam redzēs to pašu. Pakalpojumu sniedzējs, kurš saglabā satiksmes kopijas, nevarēs tās izlasīt vēlāk. Valdība, kura pavēl pakalpojuma operatoram iesniegt saturu, saņems tos pašus nesaprotamos baitus, kuri serverim bija sākotnēji.

Tas praktiskā ziņā ir ļoti daudz. Tā ir atšķirība starp vēstules rakstīšanu necaurspīdīgā aploksnē un tās rakstīšanu uz pastkartes. Abas pienāk galamērķī. Tikai viena saglabā saturu pret pastnieku.

Ko neaizsargā galašifrēšana

Ir vērts to zināt tikpat labi. E2EE neaizsargā metadatus: serveris joprojām zina, ka lietotājs A sūta datus lietotājam B, kurā stundā, cik bieži un no kurienes, pat ja tas nezina, ko viņš saka. Šie metadati, kā mēs jau apgalvojām rakstā *Šifrēt nenozīmē būt privātam*, bieži vien ir izteiksmīgāki par saturu. Zināšana, ka kāds piektdienā 22:00 trīsdesmit minūtes zvanīja uz šķiršanās advokātu biroju, stāsta stāstu, ko sarunas saturs nekad nav stāstījis. Tā ir tā pati situācija, kā redzēt personu vairākas reizes ieejam un izejam no onkoloģijas klīnikas: nav jādzird nekas no tā, par ko tiek runāts iekšpusē, lai iedomātos, kas notiek. Viens atsevišķs metadats var neko nenozīmēt; vairāki savā starpā saistīti uzzīmē kaut ko pārāk līdzīgu patiesībai. E2EE neaizsargā galapunktus: ja saņēmēja ierīci ir kompromitējusi ļaundabīga programma, ziņojums tam saņēmējam tiek atšifrēts parasti un ļaundabīgā programma to izlasa. E2EE neaizsargā pret paša sarunu biedra identitāti: ja Alise tic, ka runā ar Bruno, bet uzbrucējs ir iestarpinājies sākumā (*man in the middle*) un protokols neietver neatkarīgu pārbaudi, abas puses beidz runāt ar iebrucēju, domājot, ka runā viena ar otru.

Ir ceturtā lieta, ko ir vērts formulēt bez divdomības. E2EE neaizliedz pakalpojumu sniedzējam, kurš apgalvo, ka to piedāvā, papildus saglabāt nešifrētā ziņojuma kopiju savās sistēmās. Apgalvojums „mani ziņojumi ir galašifrēti" un apgalvojums „pakalpojumu sniedzējs nesaglabā manu saturu" nav tie paši. Lietotne var pildīt pirmo, vienlaikus pārkāpjot otro; mēs to esam redzējuši preses virsrakstos vairākkārt kopš 2018. gada. Lietotājam, ja vien klienta kods nav pārbaudāms, nav tehniska veida, kā atšķirt vienu gadījumu no otra bez ekspertu izmeklēšanas. Plašākai sabiedrībai pazīstamākais gadījums: WhatsApp ziņojumus galašifrē tranzītā, bet, ja lietotājs aktivizē dublējumu iCloud vai Google Drive bez papildu šifrēšanas, šī kopija tiek glabāta nolasāma trešās puses infrastruktūrā, un šifrēšana tiek pārtraukta paša lietotāja galā.

Jautājums, ko operators nevēlas dzirdēt

Lietotne, kura apgalvo, ka veic galašifrēšanu, tehniski var darīt vienu no trim lietām attiecībā uz atslēgām:

1. **Atslēgas atrodas tikai ierīcēs.** Tās tiek ģenerētas un atrodas tikai lietotāju ierīcēs; operators tās nezina un nesaglabā. Tas ir optimālais gadījums.

2. **Operators var piekļūt, ja vēlas.** Operatoram ir lietotāju atslēgas (vai viņš var tās ģenerēt pēc savas patikas) un tās glabā savās datu bāzēs. Ja viņš vēlas vai tiek spiests, viņš var izlasīt saturu. Tas ir vairuma „mākoņpakalpojumu" gadījums.
3. **Operators nevar piekļūt pēc konstrukcijas, bet kontrolē piekļuvi.** Operatoram nav atslēgu, bet viņš kontrolē lietotni, kas tās ģenerē. Ja viņš tiek spiests, viņš var nosūtīt ļaundabīgu atjauninājumu, kas pārtver atslēgas vai saturu pirms šifrēšanas. Tas ir daudzu komerciālo E2EE pakalpojumu gadījums.

Tāpēc operatīvais jautājums nav par to, vai kaut kas ir šifrēts, bet gan par to, kurš kontrolē ierīci un programmatūru, kas pārvalda atslēgas. Solo2 atslēgas atrodas vienīgi jūsu Seifā (ar jūsu paroli šifrēta IndexedDB), un programmatūra ir pārbaudāms atvērtais pirmkods.

Profesionālam lasītājam

Galašifrēšana ir digitālās suverenitātes rīks. Bet, tāpat kā jebkurš rīks, tā efektivitāte ir atkarīga no rokas, kas to tur, un pamata, uz kura tas balstās.

1. Kur tiek ģenerētas kriptogrāfiskās atslēgas un kur tās fiziski atrodas? Ja operators var tām piekļūt (pat īslaicīgi, pat aizbildinoties ar atgūšanu), E2EE ir tikai nominmāls.
2. Vai pastāv neatkarīga sarunu biedra pārbaude (drošības numuri, QR kodi, ārpusjoslas salīdzināšana), kas novērš starpniekuzbrukumu (man-in-the-middle) sarunas izveides laikā?
3. Vai klienta kods ir auditējams — atvērts, publicēts, reproducējams — vai arī ir jāuzticas pakalpojumu sniedzēja vārdam par to, ko klients patiešām dara?
4. Kādus metadatus pakalpojums ģenerē un saglabā, un cik ilgi? Pat ja saturs ir necaurredzams, metadati var rekonstruēt lielu daļu sensitīvās informācijas.

Šie četri jautājumi neprasa sarežģītu tehnisko informāciju; tie prasa informāciju, uz kuru jebkurš godīgs operators var atbildēt savā publiskajā dokumentācijā. Atbildes kvalitāte un precizitāte pasaka par produktu tikpat daudz kā pati atbilde.

Galašifrēšana, ja tā veikta pareizi, ir viena no smalkākajām konstrukcijām, ko mūsdienu kriptogrāfija ir sniegusi ikdienas praksei. Oriģinālā ideja — divi cilvēki var vienoties par noslēpumu publiskā kanālā — pieder Whitfield Diffie un Martin Hellman, 1976. gads; pusgadsimtu vēlāk mēs joprojām dzīvojam tās sekās. Tačiau, kā jau ar jebkuru tehnisku solījumu, tā vērtība ir atkarīga no reālā izpildījuma, nevis no etiķetes. Godīga profesionāļa jautājums nav „vai tas ir šifrēts?", bet gan „kuram ir atslēgas?". Atbildēm ir dažādas sekas. Ir vērts tās zināt.

Avoti un papildu literatūra

- Diffie, W.; Hellman, M. — *New Directions in Cryptography*, IEEE Transactions on Information Theory, 1976. gada novembris. Publiskās atslēgas kriptogrāfijas pamatraksts.
- Perrin, T.; Marlinspike, M. — *The Double Ratchet Algorithm*, publiska Open Whisper Systems specifikācija, 2016. gada pārskatījums. Signal protokola un tā industriālo atvasinājumu pamats.
- RFC 7748 — Elliptic Curves for Security (IETF, 2016. gada janvāris). Mūsdienu atslēgu apmaiņā izmantoto X25519 un X448 līkņu normatīvā specifikācija.
- Ferguson, N.; Schneier, B.; Kohno, T. — *Cryptography Engineering: Design Principles and Practical Applications* (Wiley, 2010). Nodaļas par atslēgu apmaiņu un autentificētas šifrēšanas protokoliem.
- Regula (ES) 2024/1183 par Eiropas digitālās identitātes regulējumu (eIDAS 2) — izveido ietvarus, kuros neatkarīga sarunu biedra pārbaude iegūst institucionālu atbalstu un kuros atšķirībai starp nominālo un reālo šifrēšanu ir dažādas juridiskās sekas.

← IepriekšējaisKill switch un institucionālā sagrābšanaNākamais → Biznesa modelis kā uzticības signāls

Jaunākie lasījumi

- Analīze · 2026. gada 18. maijs Patiesa vs šķietama konfidencialitāte: jautājumi, ko vērts sev uzdot
- Analīze · 2026. gada 18. maijs Self-hosting kā profesionāla prakse
- Koncepcija · 2026. gada 18. maijs 24 vārdi: kas ir kriptogrāfiskā identitāte

Paņemiet šo rakstu līdzi tur, kur jums nepieciešams.

↓ Markdown ↓ Parasts teksts ↓ PDF

Fails tiks lejupielādēts jūsu ierīcē. No turienes varat to saglabāt, importēt Solo2 vai kopīgot jebkur. Cuadernos nepieņem lēmumu par galamērķi jūsu vietā.

Vaska zīmogs · SHA-256 7862ab3da84183c7e8c4aa5eff4a8d452b2bf24810c0a41b9ec5eacd1586db94

Cuadernos Lacre · Menzuri Gestión S.L. publikācija ·
autors R.Eugenio · rediģējusi Solo2 komanda.