Xifratge d'extrem a extrem, explicat de debò

El que diuen els proveïdors quan diuen E2EE, i el que no diuen. Una explicació didàctica del mecanisme i els seus límits, sense l'embolcall publicitari.

Per entendre'ns: WhatsApp diu que els teus missatges estan xifrats d'extrem a extrem. És veritat — i no és suficient. Si la còpia de seguretat va a iCloud o Google Drive sense xifratge addicional, el xifratge es trenca al teu propi telèfon. La pregunta operativa no és si està xifrat, sinó on resideixen les claus.

El que xifrar significa, de debò

Xifrar un missatge és transformar-lo en una cosa que sembli soroll per a qualsevol que no posseeixi certa informació anomenada clau. L'operació es fa al dispositiu del que envia i, amb la clau correcta, es desfà al dispositiu del que rep. Entremig, el missatge viatja com una successió de bytes sense significat aparent. Aquesta és la idea senzilla. La resta de l'article s'ocupa dels matisos que la converteixen, segons el cas, en una garantia real o en una etiqueta de mercat.

L'adjectiu *d'extrem a extrem* —en anglès *end-to-end*, abreujat E2EE— afegeix una precisió. El xifratge no es fa perquè un servidor intermedi pugui llegir-lo i lliurar-lo. Es fa perquè només els dos extrems —el dispositiu del que envia i el dispositiu del que rep— posseeixin la clau. Qualsevol servidor pel qual el missatge passi veu el soroll, no el missatge. Aquesta és la diferència tècnica amb el xifratge *en trànsit*, on el contingut va xifrat d'un servidor al següent, però cada servidor pel qual passa el desxifra per reenvia-lo, recuperant temporalment el text en clar.

La paradoxa del secret compartit

Hi ha un problema obvi. Perquè dues persones puguin xifrar i desxifrar missatges entre si, totes dues necessiten la mateixa clau. Però, com es posen d'acord en aquesta clau si tot el que s'envien, per definició, passa per un canal on algú podria estar escoltant? Acordar la clau en el mateix canal on després la usaran sembla impossible: si l'atacant l'escolta en acordar-la, podrà desxifrar tot el posterior. Durant decennis, la criptografia clàssica va resoldre això per la via dura: les claus es lliuraven en persona, abans de començar a usar-se, en trobades físiques. Els ambaixadors carregaven amb maletins de claus cosits al folre de l'abric.

En el correu electrònic contemporani, aquesta solució no escala. Si haguéssim d'anar físicament a casa de cada persona amb qui pretenguéssim comunicar-nos de forma xifrada, no arribaríem a parlar amb ningú. La pregunta plantejada fa cinquanta anys per la comunitat criptogràfica era aquesta: és possible que dues persones que no es coneixen i que només comparteixen un canal públic acordin, en aquest mateix canal públic, un secret que ningú que escolti el canal pugui conèixer?

L'elegància de Diffie-Hellman

El 1976, dos matemàtics anomenats Whitfield Diffie i Martin Hellman van demostrar una cosa aparentment impossible: que dues persones, parlant només per un canal públic —un canal on qualsevol pot escoltar tot el que diuen—, poden posar-se d'acord en una contrasenya secreta sense que cap oient pugui descobrir-la. Sembla màgia. No ho és: és matemàtica. L'intercanvi de claus Diffie-Hellman, com es coneix des de llavors, és la base de pràcticament tota la comunicació xifrada d'internet, i mig segle d'ús intensiu i escrutini acadèmic mundial avalen la seva solidesa. Qui vulgui veure la intuïció visual o la matemàtica pot seguir llegint. Qui prefereixi confiar que funciona també pot continuar sense perdre el fil de l'article.

Per a qui vulgui intuir-ho en una imatge, hi ha una analogia coneguda amb colors. Imagina que l'Alícia i en Bru acorden en obert un color base —diguem groc— a la vista de l'Eva, que els escolta. Cadascú tria en privat un segon color secret i barreja el seu secret amb el groc. L'Alícia obté un taronja particular; en Bru obté un verd particular. Intercanvien els resultats a la vista de l'Eva. Ara cadascú barreja el color rebut amb el seu propi secret, i tots dos arriben al mateix color final, perquè l'ordre de les barreges no importa. L'Eva ha vist el groc i les dues barreges intermèdies, però no els secrets; sense algun dels secrets no pot arribar al color final. La matemàtica real canvia els colors per exponenciacions en grups modulars o corbes el·líptiques, però la idea és la mateixa: el secret compartit es construeix en públic sense que ningú al canal pugui reconstruir-lo.

En aritmètica, per a qui prefereixi veure el mecanisme: l'Alícia tria un número secret a, en Bru tria b. Intercanvien g^a i g^b en obert sobre el canal. l'Alícia calcula $(g^b)^a$ i en Bru calcula $(g^a)^b$; tots dos arriben al mateix g^{ab}. L'Eva veu g, g^a i g^b passar pel canal, però recuperar a des de g^a — l'anomenat problema del logaritme discret— requereix un temps de còmput astronòmicament superior a l'edat de l'univers quan g es tria en un grup matemàtic adequat.

Per a qui vulgui comprovar-ho amb números petits. L'intercanvi Diffie-Hellman es pot recórrer sencer amb xifres prou reduïdes com per fer els comptes a mà. Qui prefereixi no entrar en aritmètica pot saltar-se aquest bloc sense perdre el fil de l'article; qui vulgui veure el mecanisme funcionant pas a pas ho trobarà aquí. **Les regles públiques**, que qualsevol pot llegir: un primer $p = 11$ (en el Diffie-Hellman real és d'unes tres-

centes xifres; fem servir onze perquè els comptes càpiguen en una pàgina), una base $g = 2$, i la convenció que tota l'aritmètica es fa *mòdul p* — es calcula, es divideix entre p, i es conserva la resta, com un rellotge d'onze posicions que torna al zero en depassar el deu. **Les eleccions privades**, una cadascú i mai compartides: l'Alícia tria $a = 4$. En Bru tria $b = 7$.

Pas 1. L'Alícia calcula $2^4 = 16$, després $16 \bmod 11 = \mathbf{5}$. Envia el cinc. L'Eva l'anota.

Pas 2. En Bru calcula $2^7 = 128$, després $128 \bmod 11 = \mathbf{7}$. Envia el set. L'Eva també l'anota. Després dels dos enviaments, la llibreta de l'Eva conté quatre dades: $p = 11$, $g = 2$, A = 5, B = 7. Li falta el número compartit que l'Alícia i en Bru estan a punt de derivar — i que l'Eva no podrà reconstruir.

Pas 3. L'Alícia agafa el set que en Bru li va enviar i l'eleva al seu exponent privat $a = 4$. Per evitar manejar $7^4 = 2401$, es calcula per parts aplicant el mòdul en cada pas:

$7^2 = 49$

$49 \bmod 11 = 5$

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

$25 \bmod 11 = 3$

L'Alícia obté el número **3**.

Pas 4. En Bru agafa el cinc que l'Alícia li va enviar i l'eleva al seu exponent privat $b = 7$. De nou per parts:

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

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

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

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

En Bru obté també **3**.

Els dos han arribat al mateix número, 3, treballant en paral·lel. Cap d'ells no va enviar el seu exponent privat en cap moment. L'Alícia no sap que $b = 7$; en Bru no sap que $a = 4$. Cadascú va usar el valor públic que l'altre va enviar combinat amb el seu propi exponent privat, i es van trobar en la mateixa destinació. **Per què arriben al mateix número?** El que va calcular cadascun: l'Alícia, $(g^b)^a = 2^{7\times4} = 2^{28} \bmod 11$. En Bru, $(g^a)^b = 2^{4\times7} = 2^{28} \bmod 11$. És la mateixa quantitat perquè l'ordre de multiplicació d'exponents no importa ($7 \times 4 = 4 \times 7$). Cadascú va arribar per un camí diferent a la mateixa destinació.

I l'Eva? Té a la seva llibreta $p = 11$, $g = 2$, A = 5, B = 7, i voldria el 3. Per calcular-ho necessitaria conèixer a o b — però cap dels dos ha viatjat pel canal. La seva única via és preguntar-se: «per a quin exponent a es compleix $2^a \bmod 11 = 5$?». Amb un p tan petit pot provar 0, 1, 2, 3, 4... i trobar-ho en menys d'un minut. Però en substituir 11 per un primer de tres-centes xifres, l'espai d'exponents possibles té més elements que àtoms hi ha a l'univers observable. **No existeix avui dia cap algorisme conegut per la humanitat que pugui recórrer aquest espai en menys de milers de milions d'anys.** És l'anomenat *problema del logaritme discret*: fàcil cap endavant, computacionalment impossible cap enrere. I és la raó per la qual el xifratge resisteix encara que l'Eva hagi seguit tota la conversa lletra per lletra.

Tres ingredients simples —aritmètica sobre un rellotge, exponenciació, i commutativitat de la multiplicació ($a \cdot b = b \cdot a$)— combinats produeixen un protocol del qual mitja humanitat depèn cada dia per a les seves comunicacions privades. Cap de les tres peces, per separat, sembla especial. El que és decisiu és l'assemblatge.

De Diffie-Hellman al protocol Signal

El xifratge d'extrem a extrem que usen avui les aplicacions de missatgeria professional descansa, gairebé sense excepció, sobre una versió elegant i endurida de l'intercanvi Diffie-Hellman. El protocol Signal, dissenyat per Trevor Perrin i Moxie Marlinspike entre 2013 i 2016, és la referència. Combina dues idees clau. La primera, l'intercanvi de claus en corbes el·líptiques (X25519), que produeix el secret compartit inicial entre dos dispositius. La segona, l'anomenat Double Ratchet —doble engranatge—, que renova les claus automàticament amb cada missatge, de manera que comprometre el dispositiu avui no permet desxifrar missatges passats, ni missatges futurs una vegada s'ha rotat l'engranatge.

En Zig, l'intercanvi X25519 que produeix el secret compartit entre dos dispositius cap en sis línies, usant la biblioteca estàndard:

```
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)
```

El que passa en aquestes sis línies: Les claus públiques viatgen obertament. Les claus privades no surten mai del dispositiu respectiu. Cada part deriva, a partir de la seva privada i la pública de l'altra, un mateix secret de trenta-dos bytes que ningú al canal pot recuperar. Aquest secret serveix després com a llavor per xifrar els missatges intercanviats. El Double Ratchet del protocol Signal afegeix una rotació constant d'aquest material perquè el compromís d'un instant no comprometi la resta de la conversa.

I dins de `std.crypto.dh.X25519`, què hi ha exactament? No hi ha màgia oculta. Són dues funcions curtes que es poden llegir senceres a la pròpia biblioteca estàndard de Zig. La primera deriva la clau pública des de la privada — el «g^a» de l'intercanvi:

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

En el llenguatge de l'article: la clau privada es «multiplica» —en el sentit el·líptic, no en l'aritmètic elemental— pel punt base de la corba Curve25519, i el resultat se serialitza en trenta-dos bytes. L'operació `clampedMul` és la versió endurida d'aquesta multiplicació escalar: incorpora les salvaguardes que la comunitat criptogràfica va anar afegint al llarg d'anys per resistir famílies conegudes d'atacs. Dues línies de cos de funció.

La segona funció combina la teva clau privada amb la clau pública que l'altra part t'envia. És el «$(g^b)^a$» de l'intercanvi, el que produeix el secret compartit de trenta-dos bytes que cap dels dos va arribar a transmetre:

```
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();
}
```

Dues línies més. La clau pública rebuda s'interpreta com un punt sobre la corba, i es «multiplica» per la clau privada pròpia. Per la commutativitat de l'operació de corba —anàloga a la commutativitat de la multiplicació d'exponents que vam veure en l'exemple numèric—, ambdues parts acaben amb el mateix punt serialitzat: exactament el secret compartit del qual parla l'article.

Això és tot. El que en una aplicació sembla màgia és, en la realitat, dues funcions de tres línies cadascuna. La complexitat tècnica es concentra en una sola operació, `clampedMul`, que està escrita més endavant a la mateixa biblioteca estàndard, revisada durant dècades per la comunitat criptogràfica internacional, i disponible per a qualsevol que vulgui llegir-la lletra per lletra. No hi ha caixa negra ni a la nostra aplicació ni a la biblioteca estàndard de Zig. Hi ha codi obert que un humà pot entendre, triant el ritme al qual vol entrar-hi.

Què protegeix el xifratge d'extrem a extrem

El que l'E2EE protegeix bé, assumint una implementació correcta, és el contingut del missatge en trànsit. Un servidor intermedi que rebi i reenvii les dades xifrades veurà una successió de bytes inintel·ligibles. Un atacant amb accés al cable, al router, al punt d'accés wifi, veurà el mateix. Un proveïdor del servei que conservi còpies del trànsit no podrà llegir-lo a posteriori. Un Govern que ordeni a l'operador del servei lliurar el contingut rebrà els mateixos bytes inintel·ligibles que tenia el servidor en primer lloc.

Això, en termes pràctics, és molt. És la diferència entre escriure una carta dins d'un sobre opac i escriure-la en una postal. Les dues arriben. Només una preserva el contingut davant del carter.

Què no protegeix el xifratge d'extrem a extrem

Convé saber-ho igual de bé. L'E2EE no protegeix les metadades: el servidor segueix sabent que l'usuari A envia dades a l'usuari B, a quina hora, amb quina freqüència i des d'on, encara que no sàpiga què diu. Aquestes metadades, ja ho hem argumentat a *Xifrar no és ser privat*, són sovint més reveladores que el contingut. Saber que algú va trucar a un despatx d'advocats especialitzat en divorcis un divendres a les 22:00 durant trenta minuts explica una història que el contingut de la trucada mai va explicar. És la mateixa situació que veure una persona entrar i sortir diverses vegades d'una clínica oncològica: no cal sentir res del que es parla dins per imaginar el que està passant. Una sola metadada solta pot no significar res; diverses de creuades entre si dibuixen una cosa massa semblant a la veritat. L'E2EE no protegeix els extrems: si el dispositiu del receptor està compromès per un programa maliciós, el missatge es desxifra normalment per a aquest receptor i el programa maliciós el llegeix. L'E2EE no protegeix contra la identitat de l'interlocutor en si: si l'Alícia creu estar parlant amb en Bru però un atacant s'ha interposat a l'inici (un *man in the middle*) i el protocol no inclou verificació independent, les dues parts acaben parlant amb l'intrús pensant que parlen entre si.

Hi ha una quarta cosa que convé formular sense ambigüitat. L'E2EE no impedeix que un proveïdor que afirma oferir-lo guardi, a més, una còpia del missatge sense xifrar en els seus propis sistemes. L'afirmació «els meus missatges estan xifrats d'extrem a extrem» i l'afirmació «el proveïdor no conserva el meu contingut» no són la mateixa. Una aplicació pot complir la primera mentre incompleix la segona; ho hem vist en titulars de premsa repetidament des del 2018. L'usuari, tret que el codi del client sigui verificable, no té forma tècnica de distingir un cas de l'altre sense investigació experta. El cas més conegut en el públic general: WhatsApp xifra els missatges d'extrem a extrem en trànsit, però si l'usuari activa la còpia de seguretat a iCloud o Google Drive sense xifratge addicional, aquesta còpia s'emmagatzema llegible en infraestructura d'un tercer, i el xifratge es trenca a l'extrem del propi usuari.

La pregunta que l'operador no vol sentir

Una aplicació que afirma xifrar d'extrem a extrem pot, tècnicament, fer una de tres coses pel que fa a les claus:

1. **Les claus resideixen només en els dispositius.** Es generen i resideixen exclusivament en els dispositius dels usuaris; l'operador no les coneix ni les emmagatzema. És el cas òptim.
2. **L'operador pot accedir-hi si vol.** L'operador té les claus dels usuaris (o pot generar-les al seu gust) i les guarda en les seves bases de dades. Si vol o se l'obliga, pot llegir el contingut. Aquest és el cas de la majoria de serveis «al núvol».
3. **L'operador no pot accedir-hi per disseny, però controla l'accés.** L'operador no té les claus, però té el control de l'aplicació que les genera. Si se l'obliga, pot enviar una actualització maliciosa que capturi les claus o el contingut abans de xifrar. Aquest és el cas de molts serveis E2EE comercials.

La pregunta operativa, per tant, no és si una cosa està xifrada, sinó qui té el control del dispositiu i del programari que gestiona les claus. A Solo2, les claus resideixen únicament en la teva Bòveda (IndexedDB xifrada amb la teva contrasenya) i el programari és codi obert verificable.

Per al lector professional

El xifratge d'extrem a extrem és una eina de sobirania digital. Però com tota eina, la seva eficàcia depèn de la mà que l'empunya i del sòl en què es recolza.

1. On es generen les claus criptogràfiques i on resideixen físicament? Si l'operador pot accedir-hi (fins i tot temporalment, fins i tot sota formulació de recuperació), l'E2EE és nominal.
2. Existeix verificació independent de l'interlocutor (números de seguretat, codis QR, comparació fora de banda) que impedeixi un atac d'home en el medi durant l'establiment de la conversa?
3. El codi del client és auditable —obert, publicat, reproduïble— o exigeix confiar en la paraula del proveïdor sobre el que el client fa en realitat?
4. Quines metadades genera i conserva el servei, i per quant de temps? Encara que el contingut sigui opac, les metadades poden reconstruir bona part de la informació sensible.

Aquestes quatre preguntes no demanen informació tècnica avançada; demanen informació que qualsevol operador honest pot respondre en la seva documentació pública. La qualitat i precisió de la resposta diu tant del producte com la resposta mateixa.

El xifratge d'extrem a extrem, ben fet, és una de les construccions més fines que la criptografia contemporània ha lliurat a la pràctica quotidiana. La idea original —dues persones poden acordar un secret en un canal públic— pertany a Whitfield Diffie i Martin Hellman, 1976; mig segle després seguim vivint en la seva conseqüència. Però, com passa amb qualsevol promesa tècnica, el seu valor depèn del compliment real, no de l'etiqueta. La pregunta del professional honest no és «està xifrat?», sinó «qui té les claus?». Les respostes tenen conseqüències distintes. Convé saber-les.

Fonts i lectura addicional

- Diffie, W.; Hellman, M. — *New Directions in Cryptography*, IEEE Transactions on Information Theory, novembre de 1976. Article fundacional de la criptografia de clau pública.
- Perrin, T.; Marlinspike, M. — *The Double Ratchet Algorithm*, especificació pública d'Open Whisper Systems, revisió de 2016. Base del protocol Signal i els seus derivats industrials.
- RFC 7748 — Elliptic Curves for Security (IETF, gener de 2016). Especificació normativa de les corbes X25519 i X448 usades en intercanvis de clau moderns.
- Ferguson, N.; Schneier, B.; Kohno, T. — *Cryptography Engineering: Design Principles and Practical Applications* (Wiley, 2010). Capítols sobre intercanvi de claus i protocols de xifratge autenticat.
- Reglament (UE) 2024/1183 d'espai europeu d'identitat digital (eIDAS 2) — estableix marcs on la verificació independent de l'interlocutor adquireix suport institucional, i on la distinció entre xifratge nominal i xifratge real té conseqüències jurídiques diferents.

← AnteriorKill switch i la captura institucionalSegüent →El model de negoci com a senyal de confiança

Lectures recents

- Anàlisi · 18 de maig de 2026 Privadesa real vs aparent: les preguntes que convé fer-se
- Anàlisi · 18 de maig de 2026 Self-hosting com a pràctica professional
- Concepte · 18 de maig de 2026 Les 24 paraules: què és una identitat criptogràfica

Emporta't aquest article on el necessitis.

↓ Markdown ↓ Text pla ↓ PDF

L'arxiu es descarrega al teu dispositiu. Des d'allà pots guardar-lo, importar-lo a Solo2, o compartir-lo on vulguis. Cuadernos no decideix el destí per tu.

Segell de lacre · SHA-256 8d6b50ffc394d4cab8a14552d9d1ccd24ae0bad386c2a2283c45a488cbb8a484

Cuadernos Lacre · Una publicació de Menzuri Gestión S.L. ·
escrita per R.Eugenio · editada per l'equip de Solo2.

Aquest web no utilitza galetes i no carrega recursos de tercers. Utilitza un comptador anònim de visites allotjat (Umami, al nostre servidor europeu) i el mínim JavaScript necessari per als dos controls de la capçalera: tema clar o fosc, i selector d'idioma. Sense traquejadors, sense perfilat, sense compartir dades. Si vols seguir-nos: RSS.