Šifriranje od konca do konca, resnično razloženo

Kaj ponudniki pravijo, ko omenjajo E2EE, in kaj zamolčijo. Didaktična razlaga mehanizma in njegovih omejitev, brez reklamnega ovoja.

Da bomo jasni: WhatsApp pravi, da so vaša sporočila šifrirana od konca do konca. To je res — in to ni dovolj. Če varnostna kopija gre v iCloud ali Google Drive brez dodatnega šifriranja, se šifriranje prekine na vašem lastnem telefonu. Operativno vprašanje ni, ali je šifrirano, temveč kje domujejo ključi.

Kaj šifriranje zares pomeni

Šifriranje sporočila pomeni njegovo spreminjanje v nekaj, kar je videti kot šum za vsakogar, ki nima določene informacije, imenovane ključ. Operacija se izvede na napravi pošiljatelja in se s pravilnim ključem razveljavi na napravi prejemnika. Vmes sporočilo potuje kot zaporedje bajtov brez očitnega pomena. To je preprosta zamisel. Preostanek članka se ukvarja z odtenki, ki jo glede na primer spremenijo v resnično jamstvo ali le v marketinško oznako.

Pridevnik *od konca do konca* — v angleščini *end-to-end*, skrajšano E2EE — doda natančnost. Šifriranje ni izvedeno zato, da bi ga vmesni strežnik lahko prebral in dostavil. Izvedeno je tako, da imata samo oba konca — naprava pošiljatelja in naprava prejemnika — ključ. Vsak strežnik, skozi katerega potuje sporočilo, vidi šum, ne sporočila. To je tehnična razlika v primerjavi s šifriranjem *med prenosom*, kjer vsebina potuje šifrirana od enega strežnika do drugega, vendar jo vsak strežnik, skozi katerega potuje, dešifrira za nadaljnje pošiljanje, s čimer se začasno obnovi čisto besedilo.

Paradoks skupne skrivnosti

Obstaja očiten problem. Da bi dve osebi lahko med seboj šifrirali in dešifrirali sporočila, obe potrebujeta isti ključ. Toda kako se dogovorita o tem ključu, če vse, kar si pošiljata, po definiciji potuje skozi kanal, kjer bi nekdo lahko prisluškoval? Dogovarjanje o ključu po istem kanalu, kjer ga bosta pozneje uporabljala, se zdi nemogoče: če ga napadalec sliši ob dogovoru, bo lahko dešifriral vse nadaljnje. Desetletja je klasična kriptografija to reševala na težji način: ključi so bili predani osebno, pred začetkom uporabe, na fizičnih srečanjih. Veleposlaniki so nosili kovčke s ključi, všitimi v podlogo plašča.

V sodobni elektronski pošti ta rešitev ni prilagodljiva. Če bi morali fizično iti na dom vsake osebe, s katero nameravamo komunicirati šifrirano, ne bi prišli do besede z nikomer. Vprašanje, ki si ga je pred petdesetimi leti zastavila kriptografska skupnost, je bilo naslednje: ali je mogoče, da se dve osebi, ki se ne poznata in si delita le javni kanal, na tem istem javnem kanalu dogovorita o skrivnosti, ki je nihče, ki prisluškuje kanalu, ne more izvedeti?

Eleganca Diffie-Hellmana

Leta 1976 sta matematika Whitfield Diffie in Martin Hellman dokazala nekaj navidez nemogočega: da se dve osebi, ki govorita le prek javnega kanala — kanala, kjer lahko vsak sliši vse, kar povesta —, lahko dogovorita o tajnem geslu, ne da bi ga kateri koli poslušalec lahko odkril. Sliši se kot magija. Pa ni: je matematika. Izmenjava ključev Diffie-Hellman, kot je od takrat znana, je osnova za skoraj vso šifrirano komunikacijo na internetu, pol stoletja intenzivne uporabe in svetovnega akademskega nadzora pa potrjuje njeno trdnost. Kdor želi videti vizualno intuicijo ali matematiko, lahko bere naprej. Kdor raje zaupa, da deluje, lahko prav tako nadaljuje, ne da bi izgubil rdečo nit članka.

Za tiste, ki si to želijo predstavljati s sliko, obstaja znana analogija z barvami. Predstavljajte si, da se Alice in Bruno javno dogovorita o osnovni barvi — recimo rumeni — pred očmi Eve, ki jima prisluškuje. Vsak zasebno izbere drugo tajno barvo in svojo skrivnost zmeša z rumeno. Alice dobi določeno oranžno; Bruno dobi določeno zeleno. Pred očmi Eve si izmenjata rezultata. Zdaj vsak zmeša prejeto barvo s svojo skrivnostjo in oba prideta do iste končne barve, saj vrstni red mešanja ni pomemben. Eva je videla rumeno in obe vmesni mešanici, ne pa skrivnosti; brez katere koli od skrivnosti ne more priti do končne barve. Resnična matematika barve zamenja s potenciranjem v modularnih grupah ali eliptičnih krivuljah, vendar je zamisel ista: skupna skrivnost se zgradi javno, ne da bi jo kdor koli v kanalu lahko rekonstruiral.

V aritmetiki, za tiste, ki raje vidijo mehanizem: Alice izbere tajno število a, Bruno izbere b. Na kanalu si odprto izmenjata g^a in g^b. Alice izračuna $(g^b)^a$, Bruno pa $(g^a)^b$; oba prideta do istega g^{ab}. Eva vidi g, g^a in g^b potovati po kanalu, toda obnovitev a iz g^a — tako imenovani problem diskretnega logaritma — zahteva astronomski čas računanja, ki presega starost vesolja, če je g izbran v primerni matematični grupi.

Za tiste, ki to želijo preveriti z majhnimi števili. Izmenjavo Diffie-Hellman je mogoče v celoti prehoditi s številkami, ki so dovolj majhne, da se računica opravi ročno. Kdor se raje ne bi spuščal v aritmetiko, lahko ta blok preskoči, ne da bi izgubil rdečo nit članka; kdor želi videti mehanizem delovati korak za korakom, ga bo našel tukaj. **Javna pravila**, ki jih lahko prebere vsak: praštevilo $p = 11$ (v pravem Diffie-Hellmanu ima približno tristo števk; uporabljamo enajst, da računi gredo na eno stran), osnova $g = 2$ in dogovor, da se vsa aritmetika izvaja *po modulu* p — izračunaš, deliš s p in ohraniš ostanek, kot ura z enajstimi položaji, ki se po desetih vrne na ničlo. **Zasebne izbire**, po ena vsaka in nikoli deljena: Alice izbere $a = 4$. Bruno izbere $b = 7$.

Korak 1. Alice izračuna $2^4 = 16$, nato 16 mod 11 = **5**. Pošlje petico. Eva si jo zapiše.

Korak 2. Bruno izračuna $2^7 = 128$, nato 128 mod 11 = **7**. Pošlje sedmico. Tudi Eva si jo zapiše. Po dveh pošiljkah Evina beležnica vsebuje štiri podatke: $p = 11$, $g = 2$, A = 5, B = 7. Manjka ji skupno število, ki ga bosta Alice in Bruno zdaj izpeljala — in ga Eva ne bo mogla rekonstruirati.

Korak 3. Alice vzame sedmico, ki ji jo je poslal Bruno, in jo potencira s svojim zasebnim eksponentom $a = 4$. Da bi se izognili delu z $7^4 = 2401$, se izračun izvede po delih, pri čemer se modul uporabi pri vsakem koraku:

$7^2 = 49$

49 mod 11 = 5

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

25 mod 11 = 3

Alice dobi število **3**.

Korak 4. Bruno vzame petico, ki mu jo je poslala Alice, in jo potencira s svojim zasebnim eksponentom $b = 7$. Spet po delih:

$5^2 = 25$ mod 11 = 3

$5^4 = (5^2)^2 = 3^2 = 9$ mod 11 = 9

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

Končno $5^7 = 5^6 \times 5 = 5 \times 5 = 25$ mod 11 = **3**.

Tudi Bruno dobi **3**.

Oba sta prišla do istega števila, 3, medtem ko sta delala vzporedno. Nobeden ni nikoli poslal svojega zasebnega eksponenta. Alice ne ve, da je $b = 7$; Bruno ne ve, da je $a = 4$. Vsak je uporabil javno vrednost, ki jo je poslal drugi, v kombinaciji s svojim zasebnim eksponentom, in srečala sta se na istem cilju. **Zakaj prideta do istega števila?** Kaj je vsak izračunal: Alice, $(g^b)^a = 2^{7\times4} = 2^{28}$ mod 11. Bruno, $(g^a)^b = 2^{4\times7} = 2^{28}$ mod 11. Količina je ista, ker vrstni red množenja eksponentov ni pomemben ($7 \times 4 = 4 \times 7$). Vsak je po drugi poti prispel na isti cilj.

Kaj pa Eva? V beležnici ima $p = 11$, $g = 2$, A = 5, B = 7 in želela bi si 3. Da bi jo izračunala, bi morala poznati a ali b — a nobeden od njiju ni potoval po kanalu. Edina pot, ki ji preostane, je, da se vpraša: »za kateri eksponent a velja 2^a mod 11 = 5?«. Pri tako majhnem p lahko poskusi 0, 1, 2, 3, 4 ... in ga najde v manj kot minuti. Ko pa 11 zamenjamo s praštevilom s tristo števkami, ima prostor možnih eksponentov več elementov, kot je atomov v opazljivem vesolju. **Danes človeštvo ne pozna nobenega algoritma, ki bi lahko to območje prepotoval v manj kot milijardah let.** To je tako imenovani *problem diskretnega logaritma*: enostavno naprej, računsko nemogoče nazaj. In to je razlog, zakaj šifriranje zdrži, čeprav je Eva sledila celotnemu pogovoru črko za črko.

Tri preproste sestavine — aritmetika na uri, potenciranje in komutativnost množenja ($a \cdot b = b \cdot a$) — združene ustvarijo protokol, od katerega je polovica človeštva vsak dan odvisna pri svoji zasebni komunikaciji. Nobeden od teh treh delov se sam po sebi ne zdi poseben. Odločilno je sestavljanje.

Od Diffie-Hellmana do protokola Signal

Šifriranje od konca do konca, ki ga danes uporabljajo profesionalne aplikacije za sporočanje, skoraj brez izjeme temelji na elegantni in utrjeni različici izmenjave Diffie-Hellman. Referenca je protokol Signal, ki sta ga med letoma 2013 in 2016 zasnovala Trevor Perrin in Moxie Marlinspike. Združuje dve ključni zamisli. Prva je izmenjava ključev v eliptičnih krivuljah (X25519), ki ustvari začetno skupno skrivnost med dvema napravama. Druga je tako imenovani Double Ratchet — dvojna zaskočka —, ki samodejno obnavlja ključe z vsakim sporočilom, tako da kompromitacija naprave danes ne omogoča dešifriranja preteklih sporočil niti prihodnjih sporočil, ko se je zaskočka zavrtela.

V jeziku Zig izmenjava X25519, ki ustvari skupno skrivnost med dvema napravama, zavzame šest vrstic z uporabo standardne knjižnice:

```
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)
```

Kaj se zgodi v teh šestih vrsticah: Javni ključi potujejo odprto. Zasebni ključi nikoli ne zapustijo posamezne naprave. Vsaka stran iz svojega zasebnega in javnega ključa druge strani izpelje isto dvaintridesetbajtno skrivnost, ki je nihče v kanalu ne more obnoviti. Ta skrivnost pozneje služi kot seme za šifriranje izmenjanih sporočil. Double Ratchet protokola Signal doda nenehno rotacijo tega gradiva, tako da kompromitacija enega trenutka ne ogrozi preostanka pogovora.

In kaj točno se skriva znotraj `std.crypto.dh.X25519`? Nobene skrite čarovnije. Gre za dve kratki funkciji, ki ju je mogoče v celoti prebrati v sami standardni knjižnici jezika Zig. Prva izpelje javni ključ iz zasebnega — tisti »g^a« pri izmenjavi:

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

V jeziku članka: zasebni ključ se »pomnoži« — v eliptičnem in ne osnovnem aritmetičnem smislu — z osnovno točko krivulje Curve25519, rezultat pa se serializira v dvaintrideset bajtov. Operacija `clampedMul` je okrepljena različica tistega skalarnega množenja: vključuje zaščitne ukrepe, ki jih je kriptografska skupnost dodajala v preteklih letih, da bi se uprla znanim družinam napadov. Dve vrstici telesa funkcije.

Druga funkcija združi vaš zasebni ključ z javnim ključem, ki vam ga pošlje druga stran. To je »$(g^b)^a$« pri izmenjavi, ki ustvari tisto dvaintridesetbajtno skupno skrivnost, ki je ni nobeden od vaju nikoli prenesel:

```
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();
}
```

Še dve vrstici. Prejeti javni ključ se razlaga kot točka na krivulji in se »pomnoži« z lastnim zasebnim ključem. Zaradi komutativnosti operacije na krivulji — ki je analogna komutativnosti množenja eksponentov, kot smo videli v numeričnem primeru — obe strani na koncu dobita enako serializirano točko: natanko tisto skupno skrivnost, o kateri govori članek.

To je vse. Kar v aplikaciji izgleda kot magija, sta v resnici dve funkciji s po tremi vrsticami. Tehnična zapletenost je zgoščena v eni sami operaciji, `clampedMul`, ki je zapisana nekoliko nižje v isti standardni knjižnici, ki jo mednarodna kriptografska skupnost pregleduje že desetletja in je na voljo vsakomur, ki bi jo želel prebrati črko za črko. V naši aplikaciji ali v standardni knjižnici jezika Zig ni črne skrinjice. Obstaja odprtokodna koda, ki jo človek lahko razume, pri čemer si sam izbere tempo, s katerim se želi vanjo poglobiti.

Kaj šifriranje od konca do konca ščiti

Tisto, kar E2EE dobro ščiti, ob predpostavki pravilne izvedbe, je vsebina sporočila med prenosom. Vmesni strežnik, ki prejme in posreduje šifrirane podatke, bo videl zaporedje nerazumljivih bajtov. Napadalec z dostopom do kabla, usmerjevalnika ali dostopne točke wifi bo videl isto. Ponudnik storitve, ki hrani kopije prometa, ga pozneje ne bo mogel prebrati. Vlada, ki ponudniku storitve odredi izročitev vsebine, bo prejela iste nerazumljive bajte, kot jih je strežnik imel na začetku.

To je v praktičnem smislu veliko. To je razlika med pisanjem pisma v neprozorni ovojnici in pisanjem na razglednici. Obe prispeta. Le ena ohrani vsebino pred pismonošo.

Česa šifriranje od konca do konca ne ščiti

Vredno je vedeti enako dobro. E2EE ne ščiti metapodatkov: strežnik še vedno ve, da uporabnik A pošilja podatke uporabniku B, ob kateri uri, kako pogosto in od kod, čeprav ne ve, kaj pravi. Ti metapodatki, kot smo že utemeljili v *Šifriranje ni zasebnost*, so pogosto bolj zgovorni od vsebine. Vedeti, da je nekdo v petek ob 22.00 za trideset minut poklical odvetniško pisarno, specializirano za ločitve, pove zgodbo, ki je vsebina klica nikoli ni povedala. To je ista situacija kot videti osebo, ki večkrat vstopi v onkološko kliniko in izstopi iz nje: ni treba slišati ničesar o tem, o čemer se govori notri, da si predstavljate, kaj se dogaja. En sam izoliran metapodatek morda ne pomeni ničesar; več med seboj povezanih pa izriše nekaj preveč podobnega resnici. E2EE ne ščiti končnih točk: če je naprava prejemnika kompromitirana z zlonamernim programom, se sporočilo za tega prejemnika običajno dešifrira in zlonamerni program ga prebere. E2EE ne ščiti pred identiteto sogovornika samega po sebi: če Alice verjame, da se pogovarja z Brunom, vendar se je napadalec vrinil na začetku (t. i. *man in the middle*) in protokol ne vključuje neodvisnega preverjanja, obe strani na koncu govorita z vdorom, misleč, da se pogovarjata drug z drugim.

Obstaja četrta stvar, ki jo je vredno oblikovati brez dvomljivosti. E2EE ponudniku, ki trdi, da ga ponuja, ne preprečuje, da bi poleg tega hranil kopijo nešifriranega sporočila v svojih sistemih. Trditev »moja sporočila so šifrirana od konca do konca« in trditev »ponudnik ne hrani moje vsebine« nista isti. Aplikacija lahko izpolnjuje prvo, medtem ko krši drugo; to smo v časopisnih naslovih videli že večkrat od leta 2018. Uporabnik, razen če je koda odjemalca preverljiva, nima tehničnega načina, da bi brez strokovne preiskave razlikoval en primer od drugega. Najbolj znan primer v širši javnosti: WhatsApp sporočila šifrira od konca do konca med prenosom, toda če uporabnik aktivira varnostno kopiranje v iCloud ali Google Drive brez dodatnega šifriranja, se ta kopija shrani v berljivi obliki v infrastrukturi tretje osebe, šifriranje pa se prekine na koncu uporabnika samega.

Vprašanje, ki ga operater ne želi slišati

Aplikacija, ki trdi, da šifrira od konca do konca, lahko tehnično stori eno od treh stvari glede ključev:

1. **Ključi so samo v napravah.** Ustvarijo se in so izključno v napravah uporabnikov; operater jih ne pozna in ne hrani. To je optimalen primer.
2. **Operater lahko dostopa, če želi.** Operater ima ključe uporabnikov (ali jih lahko poljubno generira) in jih hrani v svojih podatkovnih bazah. Če želi ali je v to prisiljen, lahko prebere vsebino. To velja za večino storitev v »oblaku«.
3. **Operater nima dostopa po zasnovi, vendar nadzoruje dostop.** Operater nima ključev, ima pa nadzor nad aplikacijo, ki jih generira. Če je prisiljen, lahko pošlje zlonamerno posodobitev, ki zajame ključe ali vsebino pred šifriranjem. To velja za številne komercialne storitve

E2EE.

Operativno vprašanje torej ni, ali je nekaj šifrirano, temveč kdo ima nadzor nad napravo in programsko opremo, ki upravlja ključe. Pri Solo2 ključi domujejo izključno v vašem Trezorju (IndexedDB, šifrirana z vašim geslom), programska oprema pa je preverljiva odprtokodna koda.

Za profesionalnega bralca

Šifriranje od konca do konca je orodje za digitalno suverenost. Toda kot vsako orodje je njegova učinkovitost odvisna od roke, ki ga drži, in tal, na katerih stoji.

1. Kje so generirani kriptografski ključi in kje fizično obstajajo? Če operater lahko dostopa do njih (tudi začasno, tudi pod pretvezo obnovitve), je E2EE le nominalen.
2. Ali obstaja neodvisno preverjanje sogovornika (varnostne številke, kode QR, primerjava izven pasu), ki preprečuje napad man-in-the-middle med vzpostavljanjem pogovora?
3. Ali je kodo odjemalca mogoče revidirati — je odprta, objavljena, ponovljiva — ali pa zahteva zaupanje v besedo ponudnika o tem, kaj odjemalec dejansko počne?
4. Katere metapodatke storitev ustvarja in hrani ter kako dolgo? Tudi če je vsebina nepregledna, lahko metapodatki rekonstruirajo dobršen del občutljivih informacij.

Ta štiri vprašanja ne zahtevajo naprednih tehničnih informacij; zahtevajo informacije, na katere lahko vsak pošten operater odgovori v svoji javni dokumentaciji. Kakovost in natančnost odgovora o izdelku povesta toliko kot odgovor sam.

Šifriranje od konca do konca, če je izvedeno pravilno, je ena najfinejših konstrukcij, ki jih je sodobna kriptografija predala v vsakodnevno prakso. Prvotna ideja — da se dve osebi lahko dogovorita o skrivnosti prek javnega kanala — pripada Whitfield Diffieju in Martin Hellmanu iz leta 1976; pol stoletja pozneje še vedno živimo v njenih posledicah. Toda, kot pri vsaki tehnični obljubi, je njena vrednost odvisna od dejanskega izpolnjevanja, ne od oznake. Vprašanje poštenega strokovnjaka ni „ali je šifrirano?“, temveč „kdo ima ključe?“. Odgovori imajo različne posledice. Vredno jih je poznati.

Viri in nadaljnje branje

- Diffie, W.; Hellman, M. — *New Directions in Cryptography*, IEEE Transactions on Information Theory, november 1976. Temeljni članek o kriptografiji z javnim ključem.
- Perrin, T.; Marlinspike, M. — *The Double Ratchet Algorithm*, javna specifikacija Open Whisper Systems, revizija 2016. Osnova protokola Signal in njegovih industrijskih derivatov.
- RFC 7748 — Elliptic Curves for Security (IETF, januar 2016). Normativna specifikacija krivulj X25519 in X448, ki se uporabljata pri sodobnih izmenjavah ključev.
- Ferguson, N.; Schneier, B.; Kohno, T. — *Cryptography Engineering: Design Principles and Practical Applications* (Wiley, 2010). Poglavja o izmenjavi ključev in protokolih preverjenega šifriranja.
- Uredba (EU) 2024/1183 o evropskem okviru za digitalno identiteto (eIDAS 2) — vzpostavlja okvire, v katerih neodvisno preverjanje sogovornika pridobi institucionalno podporo in v katerih ima razlikovanje med nominalnim in resničnim šifriranjem različne pravne posledice.

← PrejšnjiKill switch in institucionalno zajetjeNaslednji →Poslovni model kot signal zaupanja

Zadnja branja

- Analiza · 18. maj 2026 Dejanska vs. navidezna zasebnost: vprašanja, ki si jih je smiselno zastaviti
- Analiza · 18. maj 2026 Self-hosting kot profesionalna praksa
- Koncept · 18. maj 2026 24 besed: kaj je kriptografska identiteta

Vzemite ta članek s seboj, kamor koli ga potrebujete.

↓ Markdown ↓ Navadno besedilo ↓ PDF

Datoteka bo prenesena v vašo napravo. Od tam jo lahko shranite, uvozite v Solo2 ali delite, kjer koli želite. Cuadernos ne odloča o cilju namesto vas.

Voščeni pečat · SHA-256 e85945b26783d257eba822614d460590497a23170e9a52214338e2bf35d75c8b

Cuadernos Lacre · Publikacija podjetja Menzuri Gestión S.L. ·
napisal R.Eugenio · uredila ekipa Solo2.