Koncové šifrovanie, vysvetlené naozaj

Čo poskytovatelia hovoria, keď hovoria o E2EE, a čo zamlčiavajú. Didaktické vysvetlenie mechanizmu a jeho limitov, bez reklamného obalu.

Aby sme si rozumeli: WhatsApp hovorí, že vaše správy sú šifrované od konca po koniec. Je to pravda — a to nestačí. Ak záloha putuje na iCloud alebo Google Drive bez ďalšieho šifrovania, šifrovanie sa prelamuje vo vašom vlastnom telefóne. Operatívna otázka neznie, či je to šifrované, ale kde sídlia kľúče.

Čo šifrovanie skutočne znamená

Zašifrovať správu znamená premeniť ju na niečo, čo vyzerá ako šum pre každého, kto nevlastní určitú informáciu nazývanú kľúč. Operácia sa vykonáva na zariadení odosielateľa a so správnym kľúčom sa zruší na zariadení príjemcu. Medzitým správa cestuje ako postupnosť bajtov bez zjavného významu. To je jednoduchá myšlienka. Zvyšok článku sa zaoberá nuansami, ktoré z nej v závislosti od prípadu robia skutočnú záruku alebo len marketingovú nálepku.

Prívlastok *koncové* — anglicky *end-to-end*, skrátene E2EE — dodáva presnosť. Šifrovanie sa nevykonáva preto, aby si ho mohol prečítať a doručiť sprostredkujúci server. Vykonáva sa tak, aby kľúč mali iba oba konce — zariadenie odosielateľa a zariadenie príjemcu. Akýkoľvek server, cez ktorý správa prechádza, vidí šum, nie správu. To je technický rozdiel oproti šifrovaniu *pri prenose*, kde obsah putuje zašifrovaný z jedného servera na druhý, ale každý server, cez ktorý prechádza, ho dešifruje, aby ho mohol preposlať, čím sa dočasne obnoví text v čitateľnej podobe.

Paradox zdieľaného tajomstva

Je tu zrejmý problém. Aby si dvaja ľudia mohli vzájomne šifrovať a dešifrovať správy, obaja potrebujú rovnaký kľúč. Ale ako sa na tomto kľúči dohodnú, ak všetko, čo si posielajú, z definície prechádza kanálom, kde by niekto mohol odpočúvať? Dohodnúť sa na kľúči v tom istom kanáli, kde ho neskôr budú používať, sa zdá nemožné: ak ho útočník pri dohode započuje, bude môcť dešifrovať všetko následné. Po celé desaťročia klasická kryptografia riešila toto tvrdou cestou: kľúče sa odovzdávali osobne pred začiatkom používania pri fyzických stretnutiach. Veľvyslanci nosili kufríky s kľúčmi prišité k podšívke kabáta.

V súčasnej elektronickej pošte toto riešenie nie je škálovateľné. Keby sme museli ísť fyzicky domov ku každému človeku, s ktorým mienime komunikovať šifrovane, s nikým by sme sa nestihli porozprávať. Otázka, ktorú si kryptografická komunita položila pred päťdesiatimi rokmi, znela takto: je možné, aby sa dvaja ľudia, ktorí sa nepoznajú a zdieľajú iba verejný kanál, dohodli v tomto istom verejnom kanáli na tajomstve, ktoré nikto, kto kanál odpočúva, nemôže poznať?

Elegancia Diffie-Hellman

V roku 1976 dvaja matematici menom Whitfield Diffie a Martin Hellman demonštrovali niečo zdanlivo nemožné: že dvaja ľudia, ktorí sa rozprávajú iba prostredníctvom verejného kanála — kanála, kde ktokoľvek môže počuť všetko, čo hovoria — sa môžu dohodnúť na tajnom hesle bez toho, aby ho akýkoľvek poslucháč mohol odhaliť. Znie to ako mágia. Nie je: je to matematika. Výmena kľúčov Diffie-Hellman, ako je odvtedy známa, je základom prakticky každej šifrovanej komunikácie na internete a polstoročie intenzívneho používania a celosvetového akademického skúmania potvrdzuje jej solídnosť. Kto chce vidieť vizuálnu intuíciu alebo matematiku, môže čítať ďalej. Kto radšej verí, že to funguje, môže tiež pokračovať bez toho, aby stratil niť článku.

Pre tých, ktorí si to chcú predstaviť, existuje známa analógia s farbami. Predstavte si, že sa Alica a Bruno verejne dohodnú na základnej farbe — povedzme žltej — pred očami Evy, ktorá ich počúva. Každý si v súkromí vyberie druhú tajnú farbu a zmieša svoje tajomstvo so žltou. Alica získa konkrétnu oranžovú; Bruno získa konkrétnu zelenú. Výsledky si vymenia pred očami Evy. Teraz každý zmieša obdržanú farbu so svojím vlastným tajomstvom a obaja dôjdu k rovnakej výslednej farbe, pretože na poradí miešania nezáleží. Eva videla žltú a obe medzizmesi, ale nie tajomstvá; bez niektorého z tajomstiev sa k výslednej farbe nedostane. Skutočná matematika nahrádza farby umocňovaním v modulárnych grupách alebo eliptických krivkách, ale myšlienka je rovnaká: zdieľané tajomstvo sa buduje verejne bez toho, aby ho ktokoľvek v kanáli mohol zrekonštruovať.

V aritmetike, pre tých, ktorí radšej vidia mechanizmus: Alica si vyberie tajné číslo a, Bruno si vyberie b. Vymenia si g^a a g^b otvorene cez kanál. Alica vypočíta $(g^b)^a$ a Bruno vypočíta $(g^a)^b$; obaja dôjdu k rovnakému g^{ab}. Eva vidí g, g^a a g^b prechádzať kanálom, ale získať a z g^a — takzvaný problém diskrétneho logaritmu — vyžaduje astronomický výpočtový čas presahujúci vek vesmíru, ak je g zvolené vo vhodnej matematickej grupe.

Pre tých, ktorí si to chcú overiť na malých číslach. Výmenu Diffie-Hellman je možné prejsť celú s číslami dostatočne malými na to, aby ste si ich spočítali ručne. Kto sa nechce púšťať do aritmetiky, môže tento blok preskočiť bez toho, aby stratil niť článku; kto chce vidieť mechanizmus fungovať krok za krokom, nájde ho tu. **Verejné pravidlá**, ktoré si môže prečítať každý: prvočíslo $p = 11$ (v skutočnom Diffie-Hellman má asi

tristo cifier; používame jedenásť, aby sa výpočty zmestili na jednu stranu), základ $g = 2$ a konvencia, že všetka aritmetika sa vykonáva *modulo p* — vypočíta sa, vydelí sa p a zvyšok sa zachová, ako hodiny s jedenástimi pozíciami, ktoré sa po prekročení desiatky vrátia na nulu. **Súkromné voľby**, jedna pre každého a nikdy nezdieľané: Alica si vyberie $a = 4$. Bruno si vyberie $b = 7$.

Krok 1. Alica vypočíta $2^4 = 16$, potom 16 mod 11 = **5**. Odošle päťku. Eva si to poznamená.

Krok 2. Bruno vypočíta $2^7 = 128$, potom 128 mod 11 = **7**. Odošle sedmičku. Eva si to tiež poznamená. Po dvoch odoslaniach obsahuje Evin zápisník štyri údaje: $p = 11$, $g = 2$, A = 5, B = 7. Chýba jej zdieľané číslo, ktoré sa Alica a Bruno chystajú odvodiť — a ktoré Eva nebude schopná zrekonštruovať.

Krok 3. Alica vezme sedmičku, ktorú jej poslal Bruno, a umocní ju na svoj súkromný exponent $a = 4$. Aby sme sa vyhli manipulácii so $7^4 = 2401$, počíta sa to po častiach s použitím modula v každom kroku:

$7^2 = 49$

49 mod 11 = 5

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

25 mod 11 = 3

Alica získa číslo **3**.

Krok 4. Bruno vezme päťku, ktorú mu poslala Alica, a umocní ju na svoj súkromný exponent $b = 7$. Opäť po častiach:

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

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

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

Konečne $5^7 = 5^6 \times 5 = 5 \times 5 = 25 \bmod 11 =$ **3**.

Bruno tiež získa **3**.

Obaja dospeli k rovnakému číslu, 3, pričom pracovali paralelne. Nikto z nich nikdy neposlal svoj súkromný exponent. Alica nevie, že $b = 7$; Bruno nevie, že $a = 4$. Každý použil verejnú hodnotu, ktorú poslal ten druhý, v kombinácii s vlastným súkromným exponentom a stretli sa v rovnakom cieli. **Prečo dospeli k rovnakému číslu?** Čo každý vypočítal: Alica, $(g^b)^a = 2^{7\times4} = 2^{28} \bmod 11$. Bruno, $(g^a)^b = 2^{4\times7} = 2^{28} \bmod 11$. Je to rovnaké množstvo, pretože na poradí násobenia exponentov nezáleží ($7 \times 4 = 4 \times 7$). Každý dospel inou cestou do rovnakého cieľa.

A Eva? Má vo svojom zápisníku $p = 11$, $g = 2$, A = 5, B = 7 a chcela by 3. Na výpočet by potrebovala poznať a alebo b — ale ani jedno z nich kanálom neprešlo. Jej jedinou možnosťou je položiť si otázku: «pre aký exponent a platí $2^a \bmod 11 = 5$?». S tak malým p môže vyskúšať 0, 1, 2, 3, 4... a nájsť ho za menej ako minútu. Ale keď nahradíme 11 prvočíslom o troch stoch cifier, má priestor možných exponentov viac prvkov než je atómov v pozorovateľnom vesmíre. **V dnešnej dobe neexistuje žiadny ľudstvu známy algoritmus, ktorý by dokázal tento priestor prejsť za dobu kratšiu ako miliardy rokov.** To je takzvaný *problém diskrétneho logaritmu*: ľahko dopredu, výpočtovo nemožné dozadu. A to je dôvod, prečo šifrovanie odolá, aj keď Eva sledovala celú konverzáciu písmeno po písmene.

Tri jednoduché ingrediencie — aritmetika na hodinách, umocňovanie a komutatívnosť násobenia ($a \cdot b = b \cdot a$) — po skombinovaní vytvárajú protokol, na ktorom každý deň závisí polovica ľudstva pri svojej súkromnej komunikácii. Žiadny z týchto troch prvkov sa sám o sebe nezdá byť ničím výnimočný. Rozhodujúce je ich zloženie.

Od Diffie-Hellman k protokolu Signal

Koncové šifrovanie, ktoré dnes používajú profesionálne komunikačné aplikácie, spočíva takmer bez výnimky na elegantnej a posilnenej verzii výmeny Diffie-Hellman. Referenciou je protokol Signal, ktorý navrhli Trevor Perrin a Moxie Marlinspike v rokoch 2013 až 2016. Kombinuje dve kľúčové myšlienky. Prvou je výmena kľúčov na eliptických krivkách (X25519), ktorá vytvára počiatočné zdieľané tajomstvo medzi dvoma zariadeniami. Druhou je takzvaný Double Ratchet — dvojitá západka —, ktorá automaticky obnovuje kľúče s každou správou, takže kompromitácia zariadenia dnes neumožňuje dešifrovať minulé správy, ani budúce správy po otočení západky.

V jazyku Zig sa výmena X25519, ktorá vytvára zdieľané tajomstvo medzi dvoma zariadeniami, zmestí na šesť riadkov s použitím štandardnej knižnice:

```
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)
```

Čo sa deje v týchto šiestich riadkoch: Verejné kľúče putujú otvorene. Súkromné kľúče nikdy neopustia príslušné zariadenie. Každá strana odvodí zo svojho súkromného kľúča a verejného kľúča druhej strany rovnaké tajomstvo s dĺžkou tridsaťdva bajtov, ktoré nikto v kanáli nemôže získať. Toto tajomstvo slúži neskôr ako základ pre šifrovanie vymieňaných správ. Double Ratchet protokolu Signal pridáva neustálu rotáciu tohto materiálu, takže kompromitácia jedného okamihu neohrozí zvyšok konverzácie.

A čo presne sa skrýva vo vnútri `std.crypto.dh.X25519`? Žiadna skrytá mágia. Sú to dve krátke funkcie, ktoré si môžete prečítať celé priamo v štandardnej knižnici jazyka Zig. Prvá z nich odvodzuje verejný kľúč zo súkromného — ono «g^a» z výmeny:

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

Jazykom článku: súkromný kľúč sa «vynásobí» — v eliptickom, nie základnom aritmetickom zmysle — základným bodom krivky Curve25519 a výsledok sa serializuje do tridsiatich dvoch bajtov. Operácia `clampedMul` je zosilnenou verziou tohto skalárneho násobenia: začleňuje záruky, ktoré kryptografická komunita v priebehu rokov pridávala, aby odolala známym rodinám útokov. Dva riadky tela funkcie.

Druhá funkcia kombinuje váš súkromný kľúč s verejným kľúčom, ktorý vám pošle druhá strana. To je to «$(g^b)^a$» z výmeny, ktoré vytvára tridsaťdva bajtové zdieľané tajomstvo, ktoré ani jeden z vás nikdy nepreniesol:

```
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();
}
```

Ďalšie dva riadky. Prijatý verejný kľúč je interpretovaný ako bod na krivke a je «vynásobený» vlastným súkromným kľúčom. Vďaka komutatívnosti operácie na krivke — analogickej komutatívnosti násobenia exponentov, ktorú sme videli na číselnom príklade — skončia obe strany s rovnakým serializovaným bodom: presne tým zdieľaným tajomstvom, o ktorom článok hovorí.

To je všetko. To, čo v aplikácii vyzerá ako mágia, sú v skutočnosti dve funkcie, každá s tromi riadkami. Technická zložitosť je sústredená do jedinej operácie, `clampedMul`, ktorá je napísaná ďalej v rovnakej štandardnej knižnici, desaťročia revidovaná medzinárodnou kryptografickou komunitou a dostupná komukoľvek, kto si ju chce prečítať písmeno po písmene. Neexistuje žiadna čierna skrinka v našej aplikácii ani v štandardnej knižnici jazyka Zig. Existuje open source kód, ktorému môže človek porozumieť a zvoliť si tempo, akým doň chce preniknúť.

Čo koncové šifrovanie chráni

To, čo E2EE dobre chráni, za predpokladu správnej implementácie, je obsah správy pri prenose. Sprostredkujúci server, ktorý prijíma a preposiela zašifrované dáta, uvidí sekvenciu nezrozumiteľných bajtov. Útočník s prístupom ku káblu, routeru, wifi prístupovému bodu uvidí to isté. Poskytovateľ služieb, ktorý uchováva kópie prevádzky, ju nebude môcť následne prečítať. Vláda, ktorá nariadi operátorovi služby vydať obsah, dostane rovnaké nezrozumiteľné bajty, ktoré mal server pôvodne.

To je v praktických termínoch veľa. Je to rozdiel medzi písaním listu do nepriehľadnej obálky a písaním na pohľadnicu. Obe dorazia. Iba jedna zachováva obsah pred poštárom.

Čo koncové šifrovanie nechráni

Je dobré to vedieť rovnako dobre. E2EE nechráni metadáta: server stále vie, že používateľ A posiela dáta používateľovi B, v koľko hodín, ako často a odkiaľ, aj keď nevie, čo hovoria. Tieto metadáta, ako sme už argumentovali v *Šifrovať neznamená byť v súkromí*, sú často výrečnejšie než obsah. Vedieť, že niekto volal do advokátskej kancelárie špecializovanej na rozvody v piatok o 22:00 po dobu tridsiatich minút, rozpráva príbeh, ktorý obsah hovoru nikdy nepovedal. Je to rovnaká situácia, ako keď vidíte človeka niekoľkokrát vchádzať a vychádzať z onkologickej kliniky: nemusíte počuť nič z toho, o čom sa hovorí vo vnútri, aby ste si predstavili, čo sa deje. Jediný osamotený metadaj nemusí nič znamenať; niekoľko vzájomne skrížených však vykresľuje niečo príliš podobné pravde. E2EE nechráni koncové body: ak je zariadenie príjemcu kompromitované škodlivým programom, správa sa pre tohto príjemcu normálne dešifruje a škodlivý program ju prečíta. E2EE nechráni proti identite samotného partnera: ak si Alica myslí, že sa rozpráva s Brunom, ale útočník sa vložil na začiatok (*man in the middle*) a protokol nezahŕňa nezávislé overenie, obe strany nakoniec hovoria s votrelcom a myslia si, že sa rozprávajú spolu.

Je tu štvrtá vec, ktorú je vhodné formulovať bez dvojznačnosti. E2EE nebráni poskytovateľovi, ktorý tvrdí, že ho ponúka, aby si navyše ponechal kópiu nezašifrovanej správy vo svojich vlastných systémoch. Tvrdenie „moje správy sú šifrované koncovým šifrovaním“ a tvrdenie „poskytovateľ neuchováva môj obsah“ nie sú to isté. Aplikácia môže spĺňať prvé, zatiaľ čo porušuje druhé; videli sme to v titulkoch novín opakovane od roku 2018. Užívateľ, ak nie je kód klienta overiteľný, nemá technický spôsob, ako odlíšiť jeden prípad od druhého bez odborného šetrenia. Najznámejší prípad u širokej verejnosti: WhatsApp šifruje správy koncovým šifrovaním pri prenose, ale ak si používateľ aktivuje zálohovanie na iCloud alebo Google Drive bez ďalšieho šifrovania, táto kópia sa uloží čitateľne v infraštruktúre tretej strany a šifrovanie sa na konci samotného užívateľa poruší.

Otázka, ktorú operátor nechce počuť

Aplikácia, ktorá tvrdí, že šifruje koncovým šifrovaním, môže technicky robiť jednu z troch vecí ohľadom kľúčov:

1. **Kľúče sídlia iba v zariadeniach.** Generujú sa a sídlia výhradne v zariadeniach používateľov; operátor ich nepozná ani neukladá. To je optimálny prípad.
2. **Operátor môže mať prístup, ak chce.** Operátor má kľúče používateľov (alebo ich môže vygenerovať podľa ľubovôle) a ukladá ich vo svojich databázach. Ak chce alebo je k tomu donútený, môže obsah čítať. To je prípad väčšiny „cloudových“ služieb.
3. **Operátor nemôže mať prístup zámerne, ale kontroluje prístup.** Operátor nemá kľúče, ale má kontrolu nad aplikáciou, ktorá ich generuje. Ak je k tomu donútený, môže poslať škodlivú aktualizáciu, ktorá zachytí kľúče alebo obsah pred zašifrovaním. To je prípad mnohých komerčných služieb E2EE.

Operatívna otázka teda neznie, či je niečo zašifrované, ale kto má kontrolu nad zariadením a softvérom, ktorý spravuje kľúče. V Solo2 kľúče sídlia výhradne vo vašom Trezore (IndexedDB zašifrovaná vaším heslom) a softvér je overiteľný open source.

Pre profesionálneho čitateľa

Koncové šifrovanie je nástrojom digitálnej suverenity. Ale ako každý nástroj, jeho účinnosť závisí od ruky, ktorá ho drží, a od pôdy, o ktorú sa opiera.

1. Kde sa generujú kryptografické kľúče a kde fyzicky sídlia? Ak k nim má operátor prístup (hoci dočasne, hoci pod zámienkou obnovy), E2EE je iba nominálne.
2. Existuje nezávislé overenie účastníka (bezpečnostné čísla, QR kódy, porovnanie mimo pásma), ktoré by zabránilo útoku man-in-the-middle počas nadväzovania konverzácie?
3. Je kód klienta kontrolovateľný — otvorený, publikovaný, reprodukovateľný —, alebo si vyžaduje dôveru v slovo poskytovateľa o tom, čo klient skutočne robí?
4. Aké metadáta služba generuje a uchováva a na ako dlho? Aj keď je obsah nepriehľadný, metadáta môžu zrekonštruovať veľkú časť citlivých informácií.

Tieto štyri otázky nepožadujú pokročilé technické informácie; požadujú informácie, na ktoré môže každý čestný operátor odpovedať vo svojej verejnej dokumentácii. Kvalita a presnosť odpovede vypovedá o produkte rovnako veľa ako odpoveď samotná.

Koncové šifrovanie, ak je vykonané správne, je jednou z najjemnejších konštrukcií, ktoré moderná kryptografia priniesla do každodennej praxe. Pôvodná myšlienka — že dvaja ľudia sa môžu dohodnúť na tajomstve prostredníctvom verejného kanála — pochádza od Whitfield Diffie a Martin Hellman z roku 1976; o pol storočia neskôr stále žijeme v jej dôsledkoch. Ale ako pri každom technickom prísľube, jeho hodnota závisí od skutočného plnenia, nie od nálepky. Otázka poctivého profesionála neznie „je to zašifrované?“, ale „kto má kľúče?“. Odpovede majú rôzne dôsledky. Je dobré ich poznať.

Zdroje a ďalšie čítanie

- Diffie, W.; Hellman, M. — *New Directions in Cryptography*, IEEE Transactions on Information Theory, november 1976. Zásadný článok o kryptografii s verejným kľúčom.
- Perrin, T.; Marlinspike, M. — *The Double Ratchet Algorithm*, verejná špecifikácia Open Whisper Systems, revízia 2016. Základ protokolu Signal a jeho priemyselných derivátov.
- RFC 7748 — Elliptic Curves for Security (IETF, január 2016). Normatívna špecifikácia kriviek X25519 a X448 používaných v moderných výmenách kľúčov.
- Ferguson, N.; Schneier, B.; Kohno, T. — *Cryptography Engineering: Design Principles and Practical Applications* (Wiley, 2010). Kapitoly o výmene kľúčov a protokoloch overeného šifrovania.
- Nariadenie (EÚ) 2024/1183 o európskom rámci pre digitálnu identitu (eIDAS 2) — zavádza rámce, v ktorých nezávislé overovanie účastníka získava inštitucionálnu podporu a kde má rozlišovanie medzi nominálnym a skutočným šifrovaním odlišné právne dôsledky.

← PredchádzajúciKill switch a inštitucionálne ovládnutieNasledujúci →Obchodný model ako signál dôvery

Nedávne čítanie

- Analýza · 18. mája 2026 Skutočné vs. zdanlivé súkromie: otázky, ktoré je vhodné si položiť
- Analýza · 18. mája 2026 Self-hosting ako profesionálna prax
- Koncept · 18. mája 2026 24 slov: čo je to kryptografická identita

Vezmite si tento článok tam, kam potrebujete.

↓ Markdown ↓ Čistý text ↓ PDF

Súbor sa stiahne do vášho zariadenia. Odtiaľ si ho môžete uložiť, importovať do Solo2 alebo zdieľať, kdekoľvek chcete. Cuadernos za vás o cieli nerozhoduje.

Vosková pečať · SHA-256 a1f5db2c536295b755470d1887b30960d7ba84f6cfb957d1a5cd36e9e46427d9

Cuadernos Lacre · Publikácia spoločnosti Menzuri Gestión S.L. ·
napísal R.Eugenio · edituje tím Solo2.

Tento web nepoužíva cookies a nenačítava zdroje tretích strán. Používa anonymné počítadlo návštev (Umami, na našom európskom serveri) a minimálny JavaScript potrebný pre dva ovládacie prvky v záhlaví: svetlý alebo tmavý motív a výber jazyka. Bez trackerov, bez profilovania, bez zdieľania údajov. Ak nás chcete sledovať: RSS.