# Koncové šifrování, vysvětleno doopravdy

Co poskytovatelé říkají, když mluví o E2EE, a co zamlčují. Didaktické vysvětlení mechanismu a jeho limitů, bez reklamního obalu.

**Abychom si rozuměli:** WhatsApp říká, že vaše zprávy jsou šifrovány od konce ke konci. Je to pravda — a to nestačí. Pokud záloha putuje na iCloud nebo Google Drive bez dalšího šifrování, šifrování se prolamuje ve vašem vlastním telefonu. Operativní otázka nezní, zda je to šifrované, ale kde sídlí klíče.

## Co šifrování skutečně znamená

Zašifrovat zprávu znamená přeměnit ji v něco, co vypadá jako šum pro každého, kdo nemá určitou informaci zvanou klíč. Operace se provádí na zařízení odesílatele a se správným klíčem se zvrátí na zařízení příjemce. Mezitím zpráva cestuje jako posloupnost bajtů bez zjevného významu. To je ta jednoduchá myšlenka. Zbytek článku se zabývá nuancemi, které z ní v závislosti na případu dělají skutečnou záruku nebo jen marketingovou nálepku.

Přívlastek *koncové* — anglicky *end-to-end*, zkráceně E2EE — dodává přesnost. Šifrování se neprovádí proto, aby si ho mohl přečíst a doručit zprostředkující server. Provádí se tak, aby klíč měly pouze oba konce — zařízení odesílatele a zařízení příjemce. Jakýkoli server, přes který zpráva prochází, vidí šum, nikoli zprávu. To je technický rozdíl oproti šifrování *při přenosu*, kdy obsah putuje zašifrovaný z jednoho serveru na druhý, ale každý server, přes který prochází, jej dešifruje, aby jej mohl přeposlat, čímž se dočasně obnoví text v čitelné podobě.

## Paradox sdíleného tajemství

Je tu zřejmý problém. Aby si dva lidé mohli vzájemně šifrovat a dešifrovat zprávy, potřebují oba stejný klíč. Ale jak se na tomto klíči dohodnou, když vše, co si posílají, z definice prochází kanálem, kde by mohl někdo odposlouchávat? Dohodnout se na klíči ve stejném kanálu, kde jej později budou používat, se zdá nemožné: pokud jej útočník při dohodě uslyší, bude moci dešifrovat vše následné. Klasická kryptografie to po celá desetiletí řešila tvrdou cestou: klíče se předávaly osobně před začátkem používání při fyzických setkáních. Velvyslanci nosili kufříky s klíči přišité k podšívce kabátu.

V současné elektronické poště toto řešení neškáluje. Kdybychom museli jít fyzicky domů ke každému člověku, se kterým hodláme komunikovat šifrovaně, s nikým bychom si nepromluvili. Otázka, kterou si kryptografická komunita položila před padesáti lety, zněla takto: je možné, aby se dva lidé, kteří se neznají a sdílejí pouze veřejný kanál, dohodli v tomto stejném veřejném kanálu na tajemství, které nikdo, kdo kanál odposlouchává, nemůže znát?

## Elegance Diffie-Hellman

V roce 1976 dva matematici jménem Whitfield Diffie a Martin Hellman demonstrovali něco zdánlivě nemožného: že dva lidé, kteří spolu mluví pouze prostřednictvím veřejného kanálu — kanálu, kde kdokoli může slyšet vše, co říkají — se mohou dohodnout na tajném hesle, aniž by ho jakýkoli posluchač mohl odhalit. Zní to jako magie. Není: je to matematika. Výměna klíčů Diffie-Hellman, jak je od té doby známa, je základem prakticky veškeré šifrované komunikace na internetu a půl století intenzivního používání a celosvětového akademického zkoumání potvrzuje její solidnost. Kdo chce vidět vizuální intuici nebo matematiku, může číst dál. Kdo raději věří, že to funguje, může také pokračovat, aniž by ztratil nit článku.

Pro ty, kteří si to chtějí představit, existuje známá analogie s barvami. Představte si, že se Alice a Bruno veřejně dohodnou na základní barvě — řekněme žluté — před očima Evy, která je poslouchá. Každý si v soukromí vybere druhou tajnou barvu a smíchá své tajemství se žlutou. Alice získá konkrétní oranžovou; Bruno získá konkrétní zelenou. Výsledky si vymění před očima Evy. Nyní každý smíchá obdrženou barvu se svým vlastním tajemstvím a oba dojdou ke stejné výsledné barvě, protože na pořadí míchání nezáleží. Eva viděla žlutou a obě mezisměsi, ale ne tajemství; bez některého z tajemství se k výsledné barvě nedostane. Skutečná matematika nahrazuje barvy umocňováním v modulárních grupách nebo eliptických křivkách, ale myšlenka je stejná: sdílené tajemství se buduje veřejně, aniž by ho kdokoli v kanálu mohl rekonstruovat.

**V aritmetice, pro ty, kteří raději vidí mechanismus:** Alice si vybere tajné číslo $a$, Bruno si vybere $b$. Vymění si $g^a$ a $g^b$ otevřeně přes kanál. Alice vypočítá $(g^b)^a$ a Bruno vypočítá $(g^a)^b$; oba dojdou ke stejnému $g^{ab}$. Eva vidí $g$, $g^a$ a $g^b$ procházet kanálem, ale získat $a$ z $g^a$ — takzvaný problém diskrétního logaritmu — vyžaduje astronomický výpočetní čas přesahující stáří vesmíru, pokud je $g$ zvoleno ve vhodné matematické grupě.

**Pro ty, kteří si to chtějí ověřit na malých číslech.** Výměnu Diffie-Hellman lze celou projít s čísly dostatečně malými na to, abyste si je spočítali ručně. Kdo se nechce pouštět do aritmetiky, může tento blok přeskočit, aniž by ztratil nit článku; kdo chce vidět mechanismus fungovat krok za krokem, najde ho zde. **Veřejná pravidla**, která si může přečíst každý: prvočíslo $p = 11$ (ve skutečném Diffie-Hellman má asi tři sta cifer; používáme jedenáct, aby se výpočty vešly na jednu stránku), základ $g = 2$ a konvence, že veškerá aritmetika se provádí *modulo* $p$ — vypočítá se,

vydělí se $p$ a zbytek se zachová, jako hodiny s jedenácti pozicemi, které se po překročení desítky vrátí na nulu. **Soukromé volby**, jedna pro každého a nikdy nesdílené: Alice si vybere $a = 4$. Bruno si vybere $b = 7$.

**Krok 1.** Alice vypočítá $2^4 = 16$, pak $16 \bmod 11 = \mathbf{5}$. Odešle pětku. Eva si to poznamená.

**Krok 2.** Bruno vypočítá $2^7 = 128$, pak $128 \bmod 11 = \mathbf{7}$. Odešle sedmičku. Eva si to také poznamená. Po dvou odesláních obsahuje Evin zápisník čtyři údaje: $p = 11$, $g = 2$, A = 5, B = 7. Chybí jí sdílené číslo, které se Alice a Bruno chystají odvodit — a které Eva nebude schopna zrekonstruovat.

**Krok 3.** Alice vezme sedmičku, kterou jí poslal Bruno, a umocní ji na svůj soukromý exponent $a = 4$. Abychom se vyhnuli manipulaci s $7^4 = 2401$, počítá se to po částech s použitím modula v každém kroku:

$7^2 = 49$

$49 \bmod 11 = 5$

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

$25 \bmod 11 = 3$

Alice získá číslo **3**.

**Krok 4.** Bruno vezme pětku, kterou mu poslala Alice, a umocní ji na svůj soukromý exponent $b = 7$. Opět po částech:

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

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

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

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

Bruno také získá **3**.

**Oba dospěli ke stejnému číslu, 3, přičemž pracovali paralelně.** Nikdo z nich nikdy neposlal svůj soukromý exponent. Alice neví, že $b = 7$; Bruno neví, že $a = 4$. Každý použil veřejnou hodnotu, kterou poslal ten druhý, v kombinaci s vlastním soukromým exponentem a setkali se ve stejném cíli. **Proč dospěli ke stejnému číslu?** Co každý vypočítal: Alice, $(g^b)^a = 2^{7\times4} = 2^{28} \bmod 11$. Bruno, $(g^a)^b = 2^{4\times7} = 2^{28} \bmod 11$. Je to stejné množství, protože na pořadí násobení exponentů nezáleží ($7 \times 4 = 4 \times 7$). Každý dospěl jinou cestou do stejného cíle.

**A Eva?** Má ve svém zápisníku $p = 11$, $g = 2$, A = 5, B = 7 a chtěla by 3. K výpočtu by potřebovala znát $a$ nebo $b$ — ale ani jedno z nich kanálem neprošlo. Její jedinou možností je položit si otázku: «pro jaký exponent $a$ platí $2^a \bmod 11 = 5$?». S tak malým $p$ může vyzkoušet 0, 1, 2, 3, 4... a najít ho za méně než minutu. Ale když nahradíme 11 prvočíslem o třech stech cifrách, má prostor možných exponentů více prvků než je atomů v pozorovatelném vesmíru. **V dnešní době neexistuje žádný lidstvu známý algoritmus, který by dokázal tento prostor projít za dobu kratší než miliardy let.** To je takzvaný *problém diskrétního logaritmu*: snadno dopředu, výpočetně nemožné dozadu. A to je důvod, proč šifrování odolá, i když Eva sledovala celou konverzaci písmeno po písmenu.

**Tři jednoduché ingredience** — aritmetika na hodinách, umocňování a komutativita násobení ($a \cdot b = b \cdot a$) — po zkombinování vytvářejí protokol, na kterém každý den závisí polovina lidstva při své soukromé komunikaci. Žádný ze tří prvků se sám o sobě nezdá být ničím výjimečný. Rozhodující je jejich složení.

# Od Diffie-Hellman k protokolu Signal

Koncové šifrování, které dnes používají profesionální komunikační aplikace, spočívá téměř bez výjimky na elegantní a posílené verzi výměny Diffie-Hellman. Referencí je protokol Signal, který navrhli Trevor Perrin a Moxie Marlinspike v letech 2013 až 2016. Kombinuje dvě klíčové myšlenky. První je výměna klíčů na eliptických křivkách (X25519), která vytváří počáteční sdílené tajemství mezi dvěma zařízeními. Druhou je takzvaný Double Ratchet — dvojitá ráčna —, která automaticky obnovuje klíče s každou zprávou, takže kompromitace zařízení dnes neumožňuje dešifrovat minulé zprávy, ani budoucí zprávy po otočení ráčny.

V jazyce Zig se výměna X25519, která vytváří sdílené tajemství mezi dvěma zařízeními, vejde na šest řádků s použitím standardní knihovny:

```
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)
```

**Co se děje v těchto šesti řádcích:** Veřejné klíče putují otevřeně. Soukromé klíče nikdy neopustí příslušné zařízení. Každá strana odvodí ze svého soukromého klíče a veřejného klíče druhé strany stejné tajemství o délce třiceti dvou bajtů, které nikdo v kanálu nemůže získat. Toto tajemství slouží později jako základ pro šifrování vyměňovaných zpráv. Double Ratchet protokolu Signal přidává neustálou rotaci tohoto materiálu, takže kompromitace jednoho okamžiku neohrozí zbytek konverzace.

A co přesně se skrývá uvnitř `std.crypto.dh.X25519`? Žádná skrytá magie. Jsou to dvě krátké funkce, které si můžete celé přečíst přímo ve standardní knihovně jazyka Zig. První z nich odvozuje veřejný klíč ze soukromého — ono «$g^a$» z výměny:

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

Jazykem článku: soukromý klíč se «vynásobí» — v eliptickém, nikoli základním aritmetickém smyslu — základním bodem křivky Curve25519 a výsledek se serializuje do dvaatřiceti bajtů. Operace `clampedMul` je zodolněnou verzí tohoto skalárního násobení: začleňuje záruky, které kryptografická komunita v průběhu let přidávala, aby odolala známým rodinám útoků. Dva řádky těla funkce.

Druhá funkce kombinuje váš soukromý klíč s veřejným klíčem, který vám pošle druhá strana. To je to «$(g^b)^a$» z výměny, které vytváří dvaatřicetibajtové sdílené tajemství, jež ani jeden z vás nikdy nepřenesl:

```
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();
}
```

Další dva řádky. Přijatý veřejný klíč je interpretován jako bod na křivce a je «vynásoben» vlastním soukromým klíčem. Díky komutativitě operace na křivce — analogické komutativitě násobení exponentů, kterou jsme viděli na číselném příkladu — skončí obě strany se stejným serializovaným bodem: přesně tím sdíleným tajemstvím, o kterém článek hovoří.

**To je vše.** To, co v aplikaci vypadá jako magie, jsou ve skutečnosti dvě funkce, každá o třech řádcích. Technická složitost je soustředěna do jediné operace, `clampedMul`, která je napsána dále ve stejné standardní knihovně, desetiletí revidována mezinárodní kryptografickou komunitou a dostupná komukoli, kdo si ji chce přečíst písmeno po písmenu. Neexistuje žádná černá skříňka v naší aplikaci ani ve standardní knihovně jazyka Zig. Existuje open source kód, kterému může člověk porozumět a zvolit si tempo, jakým do něj chce proniknout.

## Co koncové šifrování chrání

To, co E2EE dobře chrání, za předpokladu správné implementace, je obsah zprávy při přenosu. Zprostředkující server, který přijímá a přeposílá zašifrovaná data, uvidí sekvenci nesrozumitelných bajtů. Útočník s přístupem ke kabelu, routeru, wifi přístupovému bodu uvidí totéž. Poskytovatel služeb, který uchovává kopie provozu, jej nebude moci následně přečíst. Vláda, která nařídí operátorovi služby vydat obsah, obdrží stejné nesrozumitelné bajty, které měl server původně.

To je v praktických termínech hodně. Je to rozdíl mezi psaním dopisu do neprůhledné obálky a psaním na pohlednici. Obě dorazí. Pouze jedna zachovává obsah před pošťákem.

## Co koncové šifrování nechrání

Je dobré to vědět stejně dobře. E2EE nechrání metadata: server stále ví, že uživatel A posílá data uživateli B, v kolik hodin, jak často a odkud, i když neví, co říká. Tato metadata, jak jsme již argumentovali v *Šifrovat neznamená být v soukromí*, jsou často výmluvnější než obsah. Vědět, že někdo volal do advokátní kanceláře specializované na rozvody v pátek ve 22:00 po dobu třiceti minut, vypráví příběh, který obsah hovoru nikdy nevyprávěl. Je to stejná situace, jako když vidíte člověka několikrát vcházet a vycházet z onkologické kliniky: nemusíte slyšet nic z toho, o čem se mluví uvnitř, abyste si představili, co se děje. Jediný osamocený metadaj nemusí nic znamenat; několik vzájemně zkřížených však vykresluje něco příliš podobného pravdě. E2EE nechrání koncové body: pokud je zařízení příjemce kompromitováno škodlivým programem, zpráva se pro tohoto příjemce normálně dešifruje a škodlivý program ji přečte. E2EE nechrání proti identitě samotného partnera: pokud si Alice myslí, že mluví s Brunem, ale útočník se vložil na začátek (*man in the middle*) a protokol nezahrnuje nezávislé ověření, obě strany nakonec mluví s vetřelcem a myslí si, že mluví spolu.

Je tu čtvrtá věc, kterou je vhodné formulovat bez dvojznačnosti. E2EE nebrání poskytovateli, který tvrdí, že jej nabízí, aby si navíc ponechal kopii nezašifrované zprávy ve svých vlastních systémech. Tvrzení „moje zprávy jsou šifrovány koncovým šifrováním“ a tvrzení „poskytovatel neuchovává můj obsah“ nejsou totéž. Aplikace může splňovat první, zatímco porušuje druhé; viděli jsme to v titulcích novin opakovaně od roku 2018. Uživatel, pokud není kód klienta ověřitelný, nemá technický způsob, jak odlišit jeden případ od druhého bez odborného šetření. Nejznámější případ u široké veřejnosti: WhatsApp šifruje zprávy koncovým šifrováním při přenosu, ale pokud si uživatel aktivuje zálohování na iCloud nebo Google Drive bez dalšího šifrování, tato kopie se uloží čitelně v infrastruktuře třetí strany a šifrování se na konci samotného uživatele poruší.

## Otázka, kterou operátor nechce slyšet

Aplikace, která tvrdí, že šifruje koncovým šifrováním, může technicky dělat jednu ze tří věcí ohledně klíčů:

1. **Klíče sídlí pouze v zařízeních.** Generují se a sídlí výhradně v zařízeních uživatelů; operátor je nezná ani neukládá. To je optimální případ.

2. **Operátor může mít přístup, pokud chce.** Operátor má klíče uživatelů (nebo je může vygenerovat podle libosti) a ukládá je ve svých databázích. Pokud chce nebo je k tomu donucen, může obsah číst. To je případ většiny „cloudových“ služeb.
3. **Operátor nemůže mít přístup záměrně, ale kontroluje přístup.** Operátor nemá klíče, ale má kontrolu nad aplikací, která je generuje. Pokud je k tomu donucen, může poslat škodlivou aktualizaci, která zachytí klíče nebo obsah před zašifrováním. To je případ mnoha komerčních služeb E2EE.

Operativní otázka tedy nezní, zda je něco zašifrováno, ale kdo má kontrolu nad zařízením a softwarem, který spravuje klíče. V Solo2 klíče sídlí výhradně ve vašem Trezoru (IndexedDB zašifrovaná vaším heslem) a software je ověřitelný open source.

## Pro profesionální čtenáře

Koncové šifrování je nástrojem digitální suverenity. Ale jako každý nástroj, jeho účinnost závisí na ruce, která jej třímá, a na půdě, o kterou se opírá.

1. Kde se generují kryptografické klíče a kde fyzicky sídlí? Pokud k nim má operátor přístup (byť dočasně, byť pod záminkou obnovy), je E2EE pouze nominální.
2. Existuje nezávislé ověření účastníka (bezpečnostní čísla, QR kódy, porovnání mimo kanál), které by zabránilo útoku typu man-in-the-middle během navazování konverzace?
3. Je kód klienta auditovatelný — otevřený, publikovaný, reprodukovatelný —, nebo vyžaduje důvěru ve slovo poskytovatele o tom, co klient skutečně dělá?
4. Jaká metadata služba generuje a uchovává a na jak dlouho? I když je obsah neprůhledný, metadata mohou zrekonstruovat velkou část citlivých informací.

Tyto čtyři otázky nepožadují pokročilé technické informace; požadují informace, na které může každý poctivý operátor odpovědět ve své veřejné dokumentaci. Kvalita a přesnost odpovědi vypovídá o produktu stejně jako odpověď samotná.

---

*Koncové šifrování, pokud je provedeno správně, je jednou z nejjemnějších konstrukcí, které moderní kryptografie přinesla do každodenní praxe. Původní myšlenka — že se dva lidé mohou dohodnout na tajemství prostřednictvím veřejného kanálu — pochází od Whitfield Diffie a Martin Hellman z roku 1976; o půl století později stále žijeme v jejích důsledcích. Ale jako u každého technického slibu, jeho hodnota závisí na skutečném plnění, nikoli na nálepce. Otázka poctivého profesionála nezní „je to zašifrované?“, ale „kdo má klíče?“. Odpovědi mají různé důsledky. Je dobré je znát.*

### Zdroje a další čtení


- Diffie, W.; Hellman, M. — *New Directions in Cryptography*, IEEE Transactions on Information Theory, listopad 1976. Zásadní článek o kryptografii s veřejným klíčem.
- Perrin, T.; Marlinspike, M. — *The Double Ratchet Algorithm*, veřejná specifikace Open Whisper Systems, revize 2016. Základ protokolu Signal a jeho průmyslových derivátů.
- RFC 7748 — Elliptic Curves for Security (IETF, leden 2016). Normativní specifikace křivek X25519 a X448 používaných v moderních výměnách klíčů.
- Ferguson, N.; Schneier, B.; Kohno, T. — *Cryptography Engineering: Design Principles and Practical Applications* (Wiley, 2010). Kapitoly o výměně klíčů a protokolech ověřeného šifrování.
- Nařízení (EU) 2024/1183 o evropském prostoru digitální identity (eIDAS 2) — zavádí rámce, v nichž nezávislé ověřování účastníka získává institucionální podporu a kde má rozlišení mezi nominálním a skutečným šifrováním odlišné právní důsledky.



← PředchozíKill switch a institucionální ovládnutíDalší → Obchodní model jako signál důvěry


### Nedávné čtení


- Analýza · 18. května 2026 Skutečné vs. zdánlivé soukromí: otázky, které je vhodné si položit
- Analýza · 18. května 2026 Self-hosting jako profesionální praxe
- Koncept · 18. května 2026 24 slov: co je to kryptografická identita


Vezměte si tento článek tam, kam potřebujete.

↓ Markdown ↓ Prostý text ↓ PDF

Soubor se stáhne do vašeho zařízení. Odtud si jej můžete uložit, importovat do Solo2 nebo sdílet, kdekoli chcete. Cuadernos za vás o cíli nerozhoduje.


Pečeť · SHA-256 a91c0f21f7ccf438e1a141c4c98b09ac638f188a610f60efd9684a6a02221a00

Cuadernos Lacre · Publikace Menzuri Gestión S.L. ·
napsal R.Eugenio · rediguje tým Solo2.