24 besed: kaj je kriptografska identiteta

Kriptografska identiteta ni geslo: noben strežnik je ne hrani in je ni mogoče obnoviti. Didaktična razlaga mehanizma BIP39, zakaj natanko štiriindvajset besed in kakšna resnična teža pade na tistega, ki jih ima.

Da se razumemo: Če pozabite geslo za Gmail, vam ga Google ponastavi. Če izgubite 24 besed, ki sestavljajo kriptografsko identiteto, jih nimate koga prositi. Ne gre za to, da bi bil postopek strog — gre za to, da na drugi strani ni nikogar. Ta razlika je ključna.

Razlika med geslom in identiteto

Geslo v klasičnem modelu interneta ni uporabnikova identiteta. Je potrdilo. Uporabnik ima identiteto — ime, e-pošto, številko stranke — in da bi strežniku dokazal, da je tisti, za katerega se izdaja, predloži geslo, ki ga strežnik primerja s shranjenim odtisom. Če se odtisa ujemata, strežnik odobri sejo. Če se geslo izgubi, uporabnik ostane isti uporabnik; tisto, kar izgubi, je potrdilo, obstaja pa postopek obnove — e-pošta na registrirani naslov, varnostno vprašanje — da se to povrne.

Kriptografska identiteta deluje drugače. To niso poverilnice, ki bi jih nekdo primerjal s shranjenim odtisom; to *je* popolna matematična skrivnost sama po sebi. Ni važno, kje se nahaja — na papirju, v napravi ali celo na tujem strežniku — identiteta obstaja zaradi svoje matematike, ne zaradi tistega, ki jo potrjuje. Tu se pojavi lastnost, podobna tisti, ki smo jo videli v «Kaj dejansko je SHA-256»: lastništvo se ne dokazuje s kazanjem skrivnosti, temveč z njeno uporabo za podpisovanje. Tako ustvarjen podpis lahko vsakdo preveri z javno vrednostjo, ki je matematično izpeljana iz same skrivnosti, ne da bi mu bilo treba poznati skrivnost in brez posredovanja tretje osebe pri preverjanju. Kdor ima skrivnost, je identiteta; kdor jo izgubi, to preneha biti. Sodba je kategorična: **ni nikogar, ki bi ga lahko prosili, naj vam vrne identiteto. Ta nekdo ne obstaja, ker je sploh ni imel.**

Kaj predstavlja štiriindvajset besed

Kriptografska identiteta je običajno predstavljena z matematično skrivnostjo dolžine dvaintrideset bajtov — dvesto šestinpetdeset bitov. Številka, ki si jo je težko zapomniti in še težje brez napake prepisati. Kriptografska industrija je to težavo rešila leta 2013 z majhnim in elegantnim standardom, imenovanim BIP39: način za predstavitev teh dvesto šestinpetdesetih bitov kot zaporedje štiriindvajsetih besed, vzetih z uradnega seznama dva tisoč oseminštiridesetih besed. Aritmetika v ozadju se elegantno ujema; kdor jo želi videti podrobno, jo najde na robu.

Štetje se začne na koncu. Želimo predstaviti dvesto šestinpetdeset bitov skrivnosti z dodajanjem osmih bitov kontrolne vsote: skupaj dvesto štiriinšestdeset bitov. Če jih razdelimo na štiriindvajset besed — obvladljivo število za zapisovanje in narekovanje brez izgub — mora vsaka beseda prispevati natanko enajst bitov informacij. In enajst bitov je dve na enajsto potenco možnosti, torej dva tisoč oseminštirideset. Zato ima uradni besednjak BIP39 prav to velikost: seznam obstaja po meri težave, ne obratno.

Štetje ni dekorativno. Če nekdo pravilno prepiše triindvajset besed in se zmoti pri štiriindvajseti, bo kontrolna vsota to zaznala: programska oprema mu bo povedala »to zaporedje ni veljavno«. Če nekdo pravilno prepiše vseh štiriindvajset, bo programska oprema nedvoumno izpeljala isto identiteto. Izbira seznama besed je prav tako

premišljena: besede v besednjaku BIP39 so kratke, se med seboj razlikujejo, so brez diakritičnih znakov, izbrane tako, da zmanjšajo fonetične in pravopisne zmede. To je besednjak, zasnovan tako, da si ga ljudje zapomnijo, zapišejo in narekujejo brez izgub.

Od fraze do ključa

Teh štiriindvajset besed ni kriptografski ključ, ki podpisuje sporočila. So obnovljiva reprezentacija prvotne entropije, ki se s pomočjo deterministočnega procesa, imenovanega PBKDF2, preoblikuje v seme (seed) s štiriinšestdesetimi bajti. Iz tega semena se prav tako deterministično izpeljejo konkretni kriptografski ključi, ki jih uporabnik uporablja: zasebni ključ za podpisovanje in ustrezni javni ključ, ki se objavi za preverjanje podpisov. Isti mehanizem v različnih sistemih: kriptovalute uporabljajo krivuljo secp256k1; protokol Signal in številni sodobni sistemi uporabljajo Ed25519 na krivulji Curve25519. Za konkretno krivuljo, kot je Ed25519, standarda BIP32 in SLIP-0010 vzameta to seme s štiriinšestdesetimi bajti in deterministično izpeljeta dvaintrideset bajtov, ki tvorijo učinkovit ključ za podpisovanje — istih dvaintrideset bajtov, s katerimi se začne primer kode v naslednjem razdelku.

To je standardni način, na katerega celotna industrija predstavlja mehanizem uporabniku —denarnice za kriptovalute, upravitelji decentralizirane identitete, Signal v svojem delu za trajno identiteto, Solo2 med njimi—: uporabnik v praksi nikoli ne vidi semena ali izpeljanih ključev. Vidi štiriindvajset besed ob ustvarjanju svoje identitete in jih po želji zapiše na papir. Besede nato potujejo med njegovimi napravami, ko želi migrirati identiteto: vnese jih v novo aplikacijo, aplikacija izpelje isto seme, iste ključe, isto identiteto. To je prenosljiv, kriptografsko soliden in v mejah razumnega zapomnljiv mehanizem.

Kako se podpisuje s ključem (poteza s čopičem v Zig-u)

V Zig-u, ko imate dvaintridesetbajtno seme, izpeljano iz štiriindvajsetih besed, podpisovanje sporočila z Ed25519 zavzame le nekaj vrstic:

```
const std = @import("std");
const Ed25519 = std.crypto.sign.Ed25519;

// 'semilla' son los 32 bytes derivados de las 24 palabras.
const par = Ed25519.KeyPair.create(semilla);

// Firmar un mensaje con la clave privada:
const mensaje = "Este mensaje lo escribí yo.";
const firma = try par.sign(mensaje, null);

// Cualquiera con la clave pública del par puede verificar:
try Ed25519.Signature.verify(firma, mensaje, par.public_key);
```

Operacija podpisovanja ustvari štiriinšestdeset bajtov —imenovanih podpis— ki so se lahko generirali le iz ustreznega zasebnega ključa. Preverjanje je javno: vsakdo z javnim ključem lahko preveri, ali podpis ustreza sporočilu. Brez zasebnega ključa nihče ne more ustvariti veljavnega podpisa za to sporočilo; z javnim ključem lahko vsakdo zazna, ali je podpis veljaven. Ta asimetrija je tista, ki podpisniku omogoča dokazovanje avtorstva brez deljenja skrivnosti.

Prejšnji primer je minimalna različica iz priročnika. V dejanski kodi Solo2 veriga poteka skozi dve datoteki, eno v JavaScriptu, ki živi v uporabnikovem brskalniku in rekonstruira entropijo iz štiriindvajsetih besed, drugo v Zigu znotraj knjižnice *zcatcrypto*, ki vzame to entropijo in izpelje konkretne kriptografske ključe. Začnimo na strani brskalnika:

```
// solo2/web-app/js/lib/bip39.js
async function mnemonicToEntropy(mnemonic, lang) {
    const validation = await validateMnemonic(mnemonic, lang);
    if (!validation.valid) {
        return { entropy: null, valid: false, error: validation.error };
    }
    const wordlist = WORDLISTS[lang || 'en'];
    const words = mnemonic.trim().split(/\s+/);

    // Cada palabra aporta 11 bits (su índice en la lista de 2048).
    let bits = '';
    for (let i = 0; i < words.length; i++) {
        bits += wordlist.indexOf(words[i]).toString(2).padStart(11, '0');
    }

    // 24 palabras = 264 bits. Los primeros 256 son la entropía.
    const entropyBytes = new Uint8Array(32);
    for (let j = 0; j < 32; j++) {
        entropyBytes[j] = parseInt(bits.slice(j * 8, (j + 1) * 8), 2);
    }
    return { entropy: entropyBytes, valid: true };
}
```

Teh dvaintrideset bajtov entropije skupaj z drugimi dvaintridesetimi, izpeljanimi v istem koraku, potuje v Zigov modul WebAssembly, ki generira dejanske ključe Ed25519. Celotna funkcija s končnim čiščenjem pomnilnika se prilega na en zaslon:

```
// zcatcrypto/wasm/bindings/identity.zig
const Ed25519 = std.crypto.sign.Ed25519;
const X25519  = std.crypto.dh.X25519;

export fn identity_generate() ?*IdentityHandle {
    var seed: [64]u8 = undefined;
    if (!common.getRandomBytes(&seed)) return null;

    const handle = common.wasm_allocator.create(IdentityHandle) catch return null;

    // Bytes 0..31: semilla determinista del par Ed25519 (firma).
    const sign_kp = Ed25519.KeyPair.generateDeterministic(seed[0..32].*) catch {
        common.wasm_allocator.destroy(handle);
        return null;
    };
    handle.sign_secret = sign_kp.secret_key.toBytes();
    handle.sign_public = sign_kp.public_key.toBytes();

    // Bytes 32..63: secreto X25519 (para acordar claves de cifrado con el otro).
    handle.exchange_secret = seed[32..64].*;
    handle.exchange_public = X25519.recoverPublicKey(handle.exchange_secret) catch {
        common.wasm_allocator.destroy(handle);
        return null;
    };

    @memset(&seed, 0);  // Borra la semilla de la memoria.
```

```
    return handle;
}
```

Vredno je izpostaviti dve podrobnosti. Prva: isto seme (seed) vedno proizvede isti par ključev — prav to omogoča obnovitev identitete z vnosom štiriindvajsetih besed v novo napravo. Druga: seme se v zadnji vrstici eksplicitno izbriše iz pomnilnika. Po tej točki niti sama funkcija ne bi mogla rekonstruirati ključev; uporabnikove besede bi bile edini vir.

Za tiste, ki želijo to preveriti z majhnimi števili. Shemo podpisa je mogoče v celoti prehoditi s številkami, ki so dovolj majhne za ročni izračun. Tisti, ki raje ne bi zahajali v aritmetiko, lahko ta blok preskočijo, ne da bi izgubili rdečo nit članka; tisti, ki želijo videti mehanizem delovati korak za korakom, ga bodo našli tukaj. **Javna pravila**, ki jih lahko prebere vsak: praštevilo $p = 23$ (v dejanskem Ed25519 ima približno sedeminsedemdeset mest; uporabljamo triindvajset, da izračuni ustrezajo eni strani), osnova $g = 2$, katere red v tej skupini je $q = 11$, in dogovor, da se vsa aritmetika z g izvaja *módulo p* in se vsi eksponenti reducirajo *módulo q*. **Zasebna izbira**, ena sama in nikoli deljena: skrivnost $x = 6$. To je identiteta.

1. korak — Javni del identitete. Izračuna se enkrat in se javno objavi.

$$y = g^x \bmod p$$

$$y = 2^6 \bmod 23 = 64 \bmod 23 = 18$$

Javni del identitete je **18**. Vsakdo ga lahko vzame in uporabi za preverjanje podpisov, narejenih s to identiteto. Nihče, če opazuje le 18, ne more povrniti skrivnosti 6: to je težava diskretnega logaritma, h kateri se bomo vrnili na koncu.

2. korak — Podpisovanje sporočila. Imetnik identitete želi podpisati sporočilo $m = 7$. Začne z izbiro nove naključne vrednosti $k = 4$, ki bo uporabljena le enkrat in nikoli deljena (v dejanskem Ed25519 se k deterministično izpelje iz sporočila in skrivnosti, da se izogne nevarnosti ponovne uporabe, vendar je vloga, ki jo igra, natanko takšna). Nato izračuna tri števila:

$$r = g^k \bmod p = 2^4 \bmod 23 = 16$$

$$e = H(r, m) \bmod q = (16 + 7) \bmod 11 = 1$$

$$s = (k + x \cdot e) \bmod q = (4 + 6 \cdot 1) \bmod 11 = 10$$

Podpis je par **(r, s) = (16, 10)**. Potuje odprto skupaj s sporočilom. Vsakdo ga lahko prebere. Didaktična opomba: v dejanskem Ed25519 je funkcija H SHA-512, kriptografsko robustna; tukaj uporabljamo poenostavitev $e = (r + m) \bmod q$, da lahko bralec sledi korakom brez potrebe po izračunu zgoščene vrednosti (hash). Struktura algoritma je ista.

3. korak — Preverjanje podpisa. Preveritelj ima javni del $y = 18$, sporočilo $m = 7$ in podpis $(r, s) = (16, 10)$. Rekonstruira e na isti način — $e = (16 + 7) \bmod 11 = 1$ — in preveri, ali ta enakost drži:

$$g^s \bmod p \stackrel{?}{=} r \cdot y^e \bmod p$$

Izračuna obe strani ločeno:

Izquierda: $2^{10} \bmod 23 = 1024 \bmod 23 = 12$

Derecha: $16 \cdot 18^1 \bmod 23 = 288 \bmod 23 = 12$

Obe strani dasta **12**. Podpis je veljaven. Vsakdo z javnim delom 18 lahko pride do tega zaključka, ne da bi kdaj vedel, da je bila skrivnost 6.

Kaj pa tretja oseba, ki bi poskušala ponarejati? Eva je videla vse javno, kar gre skozi kanal: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Da bi podpisala *drugačno* sporočilo v imenu te identitete, bi morala poznati x. Njena edina pot je, da se vpraša: »za kateri eksponent x velja $2^x \bmod 23 = 18$?«. S $p = 23$ lahko poskusi 0, 1, 2, 3, … in ga najde v nekaj sekundah. Toda če 23 nadomestimo s praštevilom dejanskih dimenzij Ed25519, prostor možnih eksponentov preseže število atomov v opaznem vesolju. **Danes človeštvu ni znan noben algoritem, ki bi lahko prehodil ta prostor v manj kot milijardah let.** To je ista težava diskretnega logaritma, ki je osnova za Diffie-Hellman iz prejšnjega članka, tukaj uporabljena za shemo podpisa.

To, kar smo pravkar prehodili, je *natančno* Schnorr, shema podpisa, katere različica, prilagojena eliptični krivulji, je Ed25519. V dejanskem Ed25519 se vse operacije izvajajo na točkah določene krivulje (Curve25519) namesto na celih številih po modulu praštevila, funkcija H pa je SHA-512 namesto poenostavljene vsote, ki smo jo uporabili zgoraj. Obe zamenjavi sta izvedbeni prilagoditvi — pridobitev kriptografske odpornosti proti surovi sili, pridobitev dodatnih varnostnih lastnosti za k. Algoritmična struktura, tri operacije in razlog za asimetrijo so isti.

Tukaj je primeren kratek postanek, saj se celotna veriga na hiter pogled lahko zamenja z drugo primitivo iz tria: hashom. Ni to. Hash je edinstvena funkcija, ki stiska — vstopi veliko bajtov, izstopi kratek odtis, tam se pot konča. Kriptografska identiteta je matematično dopolnjujoč se par: skrivnost ostane in podpisuje; njen javni dvojnik se objavi in preverja. Kjer hash zruši informacije v eno smer, identiteta vzpostavi asimetrijo med dvema polovicama. Hash pričuje o tem, kaj je bilo rečeno; identiteta pričuje o tem, kdo je to rekel.

Kaj fraza ni

Potrebno je razjasniti tri pogoste zablode. Fraza ni geslo v pravem pomenu besede: ne primerja se z odtisom, shranjenim na strežniku; vnese se v uporabnikovo napravo za matematično rekonstrukcijo identitete. Fraza se ne obnavlja: če se izgubi, ni nikogar, ki bi ga lahko prosili zanjo; če se podvoji, se podvoji tudi identiteta. Fraza ni poverilnica, ki bi jo bilo mogoče ločiti od identitete: fraza *je* identiteta. Kdor jo ima, lahko deluje kot ta identiteta, brez dodatnega dovoljenja, brez postopka avtorizacije, brez možnosti obnovitve.

Ravno ta tretja lastnost spreminja težo zadeve. Izgubljeno geslo je administrativna nevšečnost. Izgubljena kriptografska identiteta je identiteta sama. Papir s frazo, ki ga najdejo tretje osebe, ni tveganje za krajo računa: to je predaja celotne identitete. Obljubo sistema — da vam nihče ne more preklicati identitete ali vas samovoljno blokirati — nerazdružljivo spremlja odgovornost — da ste vi edini varuh nečesa, česar nihče ne more obnoviti namesto vas.

Obljuba in teža

Model kriptografske identitete se običajno označuje kot *samosuverena* —self-sovereign v angleški literaturi—. Izbira besede je namerna in precej natančno opisuje stanje. Uporabnik je suveren nad svojo identiteto v skoraj srednjeveškem pomenu: ne podeljuje je noben kralj, noben izdajatelj, nobena osrednja oblast; niti je nihče od naštetih ne more odvzeti. Toda tako kot srednjeveški monarh tudi uporabnik nosi celotno posledico svojih napak: ni regenta, ki bi sprejemal odločitve namesto njega, če izgubi pečat.

Izbira med identiteto, ki jo upravlja tretja oseba, in samosuvereno identiteto nima enega univerzalnega pravilnega odgovora. Za račun na nepomembnem forumu je upravljana identiteta verjetno sorazmerna s tveganjem. Za poklicno identiteto, ki podpisuje pravno zavezujoče dokumente, za ekonomsko identiteto, ki varuje lastne prihranke, za identiteto poklicne komunikacije s strankami, ki so zaupale občutljive informacije, se vprašanje spremeni. Tam vprašanje preneha biti „ali je udobno?“ in postane „kdo ima poleg mene moč delovati kot jaz in v kakšnih okoliščinah?“.

Kje se ta mehanizem pojavi v dejanskih sistemih

BIP39 se je rodil v svetu Bitcoina leta 2013 in se hitro razširil na celoten ekosistem kriptovalut: vsaka resna denarnica danes sprejme dvanajst- ali štiriindvajsetbesedno frazo BIP39 kot varnostno kopijo ekonomske identitete svojega imetnika. Zunaj kriptovalut se isti osnovni koncept — kriptografski par, ki dokazuje avtorstvo brez posrednika — pojavlja v drugih sistemih z drugačno sintakso. Ključi SSH, ki jih sistemski administrator uporablja za dostop do svojih strežnikov, so klasičen primer: zasebni ključ, ki ga administrator hrani na svojem računalniku, in javni, ki se kopira na vsak strežnik; noben subjekt, primerljiv s centralizirano storitvijo, ne posreduje. Protokol Signal uporablja Ed25519 s perzistentnim materialom ključa na napravi; evropski eIDAS se v svojem delu o kvalificiranem podpisu opira na isto kriptografsko načelo, s to razliko, da ključ hrani kvalificiran ponudnik storitev zaupanja namesto uporabnika.

Solo2, založniška platforma te publikacije, uporablja štiriindvajsetbesedno frazo BIP39 kot identiteto vsakega uporabnika. Uporabnik ob ustvarjanju računa besede vidi enkrat. Ne shranjujejo se na nobenem strežniku Solo2 ali kogar koli drugega: če si jih uporabnik zapiše in jih hrani, svojo identiteto ohrani za vedno. Če jih izgubi, jih izgubi. To je logična posledica arhitekture brez posrednika: če bi Solo2 lahko vrnil identiteto uporabniku, ki jo je izgubil, bi jo lahko dal tudi komur koli, ki bi pritisnil na Solo2, da mu jo izroči.

Za profesionalnega bralca

Štirje premisleki za tiste, ki ocenjujejo uvedbo kriptografske samostojne (autosoberana) identitete v profesionalnem kontekstu:

1. Fraza je identiteta. Fizično varovanje — papir, več kopij na različnih mestih, na koncu vgravirana kovina za dolgotrajno uporabo — ponuja več jamstev kot digitalno varovanje, ki povečuje napadalno površino, ne da bi zmanjšalo tveganje izgube.
2. Obnove ni. Načrtovanje procesa ob predpostavki, da bo nekega dne primarna kopija izgubljena, je veliko bolj modro kot odkritje tega na dan izgube. Druga geografsko ločena kopija reši skoraj vse scenarije.
3. To ni isto kot kvalificirano potrdilo eIDAS. Za kvalificiran podpis v Uniji — notarske listine, določeni postopki z upravo — zakonodaja zahteva kvalificiranega ponudnika, ki hrani ključ. Kriptografska samostojna identiteta služi profesionalni komunikaciji in dokumentarnemu podpisovanju z dokazno vrednostjo, vendar ne nadomešča samodejno kvalificiranega potrdila v primerih, ko norma to zahteva.
4. Če se bo identiteta prenašala — dedovanje, profesionalno nasledstvo, prenehanje dejavnosti — je priporočljivo postopek pripraviti prej, ne pozneje. Formalni postopki z ovojnicami, zapečatenimi s pečatnim voskom (lacre), navodila izvršitelju oporoke, polog v notarski pisarni, so klasični dogovori, ki so popolnoma združljivi s kriptografsko naravo sredstva.

Ta članek zaključuje konceptualni trio, ki je odprl cikel — hash, šifriranje, identiteta —. Tri ideje se gradijo ena na drugi: hash daje nespremenljiv odtis, šifriranje daje zaupnost brez zaupanja vredne tretje osebe, identiteta daje avtorstvo brez podelitvene tretje osebe. Vsi trije si delijo lastnost, ki prav tako ni ideološka: prenašajo s tistega, ki upravlja storitev, na tistega, ki jo uporablja, tehnične sposobnosti, ki so tradicionalno pripadale operaterju. Z njimi prenašajo tudi odgovornosti. Iskren pogovor o katerem koli od teh treh zahteva pogovor tudi o preostalih dveh.

Viri in nadaljnje branje

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, predlog izboljšave Bitcoina iz leta 2013. De facto standard za obnovitvene fraze v kripto industriji.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), vključno z Ed25519. IETF, januar 2017. Normativna specifikacija sheme podpisa, ki se uporablja v velikem delu sodobne industrije.
- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, različica 2.0. IETF, september 2000. Določa algoritem PBKDF2, ki se uporablja v izpeljavi BIP39 iz fraze v seed.

- Uredba (EU) 910/2014 (eIDAS) in njen razvoj z Uredbo (EU) 2024/1183 (eIDAS 2) — evropski okvir za elektronsko identiteto in kvalificiran podpis. Režim, drugačen od samostojnega, vendar konceptualno podprt z istimi kriptografskimi primitivi.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Kanonično besedilo o načelih in zavezah samostojnega modela, zgodnejše, vendar pomembno za razumevanje družine sodobnih rešitev.

← PrejšnjiPoslovni model kot signal zaupanjaNaslednji →Self-hosting kot profesionalna praksa

Zadnja branja

- Razmislek · 29. junij 2026 Nisi anonimen
- Razmislek · 27. maj 2026 Česa podpis ne more popraviti
- Analiza · 26. maj 2026 Dejanska vs. navidezna zasebnost: vprašanja, ki si jih je smiselno zastaviti

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 0966b0a359e17ce8c23ebefff5c799f45c747270ca1ece0c0aef8cd9b319d910

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