24 vārdi: kas ir kriptogrāfiskā identitāte

Kriptogrāfiskā identitāte nav parole: neviens serveris to neglabā un tā nav atgūstama. BIP39 mehānisma didaktisks skaidrojums, kāpēc tieši divdesmit četri vārdi un kāda reāla atbildība gulstas uz to, kurš tos pārvalda.

Lai mēs saprastos: ja aizmirstat Gmail paroli, Google to atiestata. Ja pazaudējat 24 vārdus, kas veido kriptogrāfisko identitāti, nav neviena, kam tos lūgt. Nav tā, ka procedūra būtu stingra – vienkārši otrā galā neviena nav. Šī atšķirība ir pati būtība.

Atšķirība starp paroli un identitāti

Parole klasiskajā interneta modelī nav lietotāja identitāte. Tas ir apliecinājums. Lietotājam ir identitāte – vārds, e-pasts, klienta numurs – un, lai pierādītu serverim, ka viņš ir tas, par ko uzdodas, viņš uzrāda paroli, kuru serveris salīdzina ar saglabāto nospiedumu. Ja nospiedumi sakrīt, serveris atļauj sesiju. Ja parole tiek pazaudēta, lietotājs joprojām ir tas pats lietotājs; tas, ko viņš pazaudē, ir apliecinājums, un pastāv atkopšanas procedūra – e-pasts uz reģistrēto adresi, drošības jautājums – tā atjaunošanai.

Kriptogrāfiskā identitāte darbojas citādi. Tas nav akreditācijas datu kopums, ko kāds salīdzina ar saglabātu nospiedumu; tas *ir* pilnīgs matemātisks noslēpums pats par sevi. Nav nozīmes tam, kur tas atrodas – uz papīra, ierīcē vai pat svešā serverī: identitāte eksistē savas matemātikas dēļ, nevis tāpēc, kurš to apstiprina. Šeit parādās īpašība, kas līdzīga tai, ko redzējām rakstā „Kas ir SHA-256 patiesībā”: piederība netiek pierādīta, uzrādot noslēpumu, bet gan izmantojot to parakstīšanai. Šādi izveidotu parakstu ikviens var pārbaudīt ar publisku vērtību, kas matemātiski atvasināta no paša noslēpuma, nav nepieciešams zināt pašu noslēpumu un nav nepieciešama trešās puses starpniecība pārbaudē. Kas pieder noslēpums, tas ir identitāte; kas to pazaudē, tas pārstāj tāds būt. Spriedums ir kategorisks: **nav neviena, kam lūgt atdot identitāti. Šāda persona neeksistē, jo tā viņam sākotnēji nemaz nepiederēja.**

Ko reprezentē divdesmit četri vārdi

Kriptogrāfisko identitāti parasti reprezentē trīsdesmit divu baitu – divu simtu piecdesmit sešu bitu – matemātisks noslēpums. Skaitlis, kuru grūti atcerēties un vēl grūtāk bez kļūdām norakstīt. Kriptogrāfijas nozare šo problēmu atrisināja 2013. gadā ar nelielu un elegantu standartu, ko sauc par BIP39: veidu, kā reprezentēt šos divus simtus piecdesmit sešus bitus kā divdesmit četru vārdu secību, kas izvēlēta no oficiāla divu tūkstošu četrdesmit astoņu vārdu saraksta. Aritmētika aiz tā der eleganti; kas vēlas to redzēt detalizēti, atradīs to apmales piezīmē.

Skaitīšana sākas no beigām. Mēs vēlamies reprezentēt noslēpuma divus simtus piecdesmit sešus bitus, pievienojot astoņus kontrolsummas bitus: kopā divus simtus sešdesmit četrus bitus. Ja tos sadalām divdesmit četros vārdos – pārvaldāmā skaitā, ko var pierakstīt un nodiktēt bez zudumiem –, katram vārdam ir jānodrošina tieši vienpadsmit informācijas biti. Un vienpadsmit biti ir divi vienpadsmitajā pakāpē iespējas, t.i., divi tūkstoši četrdesmit astoņi. Tāpēc oficiālā BIP39 vārdnīca ir tieši šāda izmēra: saraksts izveidots atbilstoši problēmas mērogam, nevis otrādi.

Skaitīšanai nav dekoratīvas nozīmes. Ja kāds pareizi noraksta divdesmit trīs vārdus un kļūdās divdesmit ceturtajā, kontrolsumma to pamanīs: programmatūra viņam pateiks „šī secība nav derīga”. Ja kāds pareizi

noraksta visus divdesmit četrus vārdus, programmatūra viennozīmīgi atvasinās to pašu identitāti. Vārdu saraksta izvēle arī ir mērķtiecīga: BIP39 vārdnīcas vārdi ir īsi, savstarpēji atšķirīgi, bez diakritiskajām zīmēm, izvēlēti, lai minimizētu fonētiskos un pareizrakstības pārpratumus. Tā ir vārdnīca, kas izstrādāta tā, lai cilvēki to varētu atcerēties, pierakstīt un nodiktēt bez zudumiem.

No frāzes līdz atslēgai

Divdesmit četri vārdi nav kriptogrāfiskā atslēga, kas paraksta ziņojumus. Tie ir atjaunojams sākotnējās entropijas attēlojums, kas, izmantojot deterministisku procesu, ko sauc par PBKDF2, tiek pārveidots par sešdesmit četru baitu sēklu (seed). No šīs sēklas arī deterministiski tiek atvasinātas konkrētās kriptogrāfiskās atslēgas, kuras lietotājs izmanto: privātā atslēga parakstīšanai un atbilstošā publiskā atslēga, kas tiek publicēta parakstu pārbaudei. Tas pats mehānisms dažādās sistēmās: kriptovalūtas izmanto secp256k1 līkni; Signal protokols un daudzas modernas sistēmas izmanto Ed25519 uz Curve25519 līknes. Konkrētai līknei, piemēram, Ed25519, BIP32 un SLIP-0010 standarti ņem šo sešdesmit četru baitu sēklu un deterministiski atvasina trīsdesmit divus baitus, kas veido faktisko parakstīšanas atslēgu — tos pašus trīsdesmit divus baitus, ar kuriem sākas koda piemērs nākamajā sadaļā.

Šis ir standarta veids, kā visa nozare iepazīstina lietotāju ar mehānismu —kriptovalūtu maki, decentralizētās identitātes pārvaldnieki, Signal tā pastāvīgās identitātes daļā, Solo2 starp tiem—: lietotājs praksē nekad neredz sēklu vai atvasinātās atslēgas. Izveidojot savu identitāti, viņš redz divdesmit četrus vārdus un pēc izvēles pieraksta tos uz papīra. Pēc tam vārdi ceļo starp viņa ierīcēm, kad viņš vēlas migrēt identitāti: viņš tos ievada jaunajā lietojumprogrammā, lietojumprogramma atvasina to pašu sēklu, tās pašas atslēgas, to pašu identitāti. Tas ir portatīvs, kriptogrāfiski pamatīgs un saprātīgās robežās iegaumējams mehānisms.

Kā parakstīt ar atslēgu (Zig triepiens)

Zig valodā, kad jums ir trīsdesmit divu baitu sēkla, kas atvasināta no divdesmit četriem vārdiem, ziņojuma parakstīšana ar Ed25519 aizņem tikai dažas rindiņas:

```
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);
```

Parakstīšanas darbība rada sešdesmit četrus baitus —sauktus par parakstu—, kurus varēja ģenerēt tikai no atbilstošās privātās atslēgas. Verifikācija ir publiska: jebkurš, kam ir publiskā atslēga, var pārbaudīt, vai paraksts atbilst ziņojumam. Bez privātās atslēgas neviens nevar izveidot derīgu parakstu šim ziņojumam; ar publisko atslēgu ikviens var noteikt, vai paraksts ir derīgs. Šī asimetrija ir tā, kas ļauj parakstītājam pierādīt autorību, neizpaužot noslēpumu.

Iepriekšējais piemērs ir rokasgrāmatas minimālā versija. Solo2 reālajā kodā ķēde iet caur diviem failiem: vienu JavaScript valodā, kas atrodas lietotāja pārlūkprogrammā un rekonstruē entropiju no divdesmit četriem vārdiem, otru Zig valodā *zcatcrypto* bibliotēkā, kas paņem šo entropiju un atvasina konkrētās kriptogrāfiskās atslēgas. Sākot no pārlūkprogrammas puses:

```
// 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 };
}
```

Šie trīsdesmit divi entropijas baiti kopā ar citiem trīsdesmit diviem, kas atvasināti tajā pašā solī, ceļo uz Zig WebAssembly moduli, kas ģenerē pašas Ed25519 atslēgas. Pilna funkcija ar tās galīgo atmiņas tīrīšanu ietilpst vienā ekrānā:

```
// 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;
}
```

Ir vērts atzīmēt divas detaļas. Pirmā: viena un tā pati sākuma vērtība (seed) vienmēr rada vienu un to pašu atslēgu pāri — tieši tas ļauj atgūt identitāti, ievadot divdesmit četrus vārdus jaunā ierīcē. Otrā: sākuma vērtība tiek skaidri izdzēsta no atmiņas pēdējā rindā. Pēc šī punkta pat pati funkcija nevarētu rekonstruēt atslēgas; lietotāja vārdi būtu vienīgais avots.

Tiem, kas vēlas to pārbaudīt ar maziem skaitļiem. Parakstīšanas shēmu var pilnībā iziet ar skaitļiem, kas ir pietiekami mazi, lai aprēķinus veiktu ar roku. Tie, kuri nevēlas iedziļināties aritmētikā, var izlaist šo bloku, nezaudējot raksta jēgu; tie, kuri vēlas redzēt mehānismu darbojamies soli pa solim, atradīs to šeit. **Publiskie noteikumi**, kurus var izlasīt ikviens: pirmskaitlis $p = 23$ (reālajā Ed25519 tas ir aptuveni septiņdesmit septiņu ciparu garš; mēs izmantojam divdesmit trīs, lai aprēķini ietilptu vienā lappusē), bāze $g = 2$, kuras kārta šajā grupā ir $q = 11$, un konvencija, ka visa aritmētika ar g tiek veikta *módulo p* un visi kāpinātāji tiek reducēti *módulo q*. **Privātā izvēle**, viena vienīga un nekad neizpausta: noslēpums $x = 6$. Tā ir identitāte.

1. solis — Identitātes publiskā daļa. Tā tiek aprēķināta vienu reizi un atklāti publicēta.

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

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

Identitātes publiskā daļa ir **18**. Ikviens var to paņemt un izmantot, lai pārbaudītu ar šo identitāti veiktos parakstus. Neviens, novērojot tikai skaitli 18, nevar atgūt noslēpumu 6: tā ir diskrētā logaritma problēma, pie kuras mēs atgriezīsimies beigās.

2. solis — Ziņojuma parakstīšana. Identitātes īpašnieks vēlas parakstīt ziņojumu $m = 7$. Viņš sāk, izvēloties jaunu nejaušu vērtību $k = 4$, kas tiks izmantota tikai vienu reizi un nekad netiks izpausta (reālajā Ed25519 k tiek atvasināts deterministiski no ziņojuma un noslēpuma, lai izvairītos no atkārtotas izmantošanas briesmām, bet tā loma ir tieši šāda). Pēc tam viņš aprēķina trīs skaitļus:

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

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

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

Paraksts ir pāris **(r, s) = (16, 10)**. Tas ceļo atklāti kopā ar ziņojumu. Ikviens var to izlasīt. Didaktiska piezīme: reālajā Ed25519 funkcija H ir SHA-512, kas ir kriptogrāfiski droša; šeit mēs izmantojam vienkāršojumu $e = (r + m) \bmod q$, lai lasītājs varētu iziet soļus bez nepieciešamības aprēķināt jaucējkodu (hash). Algoritma struktūra ir tāda pati.

3. solis — Paraksta pārbaude. Pārbaudītājam ir publiskā daļa $y = 18$, ziņojums $m = 7$ un paraksts $(r, s) = (16, 10)$. Viņš rekonstruē e tādā pašā veidā — $e = (16 + 7) \bmod 11 = 1$ — un pārbauda, vai šī vienādība izpildās:

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

Aprēķina abas puses atsevišķi:

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

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

Abas puses dod **12**. Paraksts ir derīgs. Ikviens ar publisko daļu 18 var nonākt pie šī secinājuma, nekad nezinot, ka noslēpums bija 6.

Un kā ar trešo pusi, kas mēģinātu viltot? Eva ir redzējusi visu publisko informāciju caur kanālu: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Lai parakstītu *citādu* ziņojumu šīs identitātes vārdā, viņai būtu jāzina x. Viņas vienīgais ceļš ir pajautāt sev: "kādam kāpinātājam x izpildās $2^x \bmod 23 = 18$?". Ar $p = 23$ viņa var izmēģināt 0, 1, 2, 3, … un atrast to dažu sekunžu laikā. Bet, aizstājot 23 ar pirmskaitli no reālajiem Ed25519 izmēriem, iespējamo kāpinātāju telpa pārsniedz atomu skaitu novērojamajā visumā. **Mūsdienās cilvēcei nav zināms neviens algoritms, kas varētu iziet šo telpu mazāk nekā miljardos gadu.** Tā ir tā pati diskrētā logaritma problēma, kas ir pamatā Diffie-Hellman no iepriekšējā raksta, šeit piemērota parakstīšanas shēmai.

Tas, ko mēs nupat izgājām, ir *tieši* Schnorr, parakstīšanas shēma, kuras variants ir eliptiskajai līknei pielāgotais Ed25519. Reālajā Ed25519 visas operācijas tiek veiktas ar konkrētas līknes (Curve25519) punktiem, nevis veseliem skaitļiem modulo pirmskaitlis, un funkcija H ir SHA-512 iepriekš izmantotās vienkāršotās summas vietā. Abas aizstāšanas ir ieviešanas pielāgojumi — kriptogrāfiskās izturības iegūšana pret brutālo spēku, papildu drošības īpašību iegūšana k vērtībai. Algoritmiskā struktūra, trīs operācijas un asimetrijas iemesls ir tie paši.

Šeit ir vietā īsa pauze, jo visu ķēdi no pirmā acu uzmetiena var sajaukt ar citu trijotnes primitīvu: jaucējkodu (hash). Tas tā nav. Jaucējkods ir unikāla funkcija, kas saspiež — ieiet daudz baitu, iziet īss nospiedums, tur ceļš beidzas. Kriptogrāfiskā identitāte ir matemātiski papildinošs pāris: noslēpums paliek un paraksta; tā publiskā daļa tiek publicēta un pārbaudīta. Tur, kur jaucējkods sakļauj informāciju vienā virzienā, identitāte izveido asimetriju starp divām pusēm. Jaucējkods apliecina, kas tika pateikts; identitāte apliecina, kas to pateica.

Kas frāze nav

Būtu jāizkliedē trīs bieži sastopami pārpratumi. Frāze nav parole šī vārda tiešajā nozīmē: tā netiek salīdzināta ar serverī saglabātu pirkstu nospiedumu; tā tiek ievadīta lietotāja ierīcē, lai matemātiski rekonstruētu identitāti. Frāze netiek atjaunota: ja tā tiek pazaudēta, nav neviena, kam to lūgt; ja tā tiek dublēta, tiek dublēta arī identitāte. Frāze nav no identitātes atdalāms akreditācijas datu kopums: frāze *ir* identitāte. Tas, kuram tā pieder, var rīkoties tās vārdā bez papildu atļaujas, bez autorizācijas procesa, bez atjaunošanas iespējas.

Tieši šī trešā īpašība ir tā, kas maina lietas svarīgumu. Pazaudēta parole ir administratīvs apgrūtinājums. Pazaudēta kriptogrāfiskā identitāte ir pati identitāte. Trešo pušu atrasts papīrs ar frāzi nav tikai konta zādzības risks: tā ir visas identitātes nodošana. Sistēmas solījums — ka neviens nevar atsaukt jūsu identitāti vai patvaļīgi jūs bloķēt — ir neatraujami saistīts ar atbildību — ka jūs esat vienīgais sargs kaut kam, ko neviens nevar atjaunot jūsu vietā.

Solījums un svars

Kriptogrāfiskās identitātes modeli bieži sauc par *pašsuverēnu* —self-sovereign angļu valodas literatūrā—. Vārda izvēle ir apzināta un diezgan precīzi raksturo stāvokli. Lietotājs ir suverēns pār savu identitāti gandrīz viduslaiku nozīmē: to nepiešķir neviens karalis, neviens emitents, neviena centrālā iestāde; tāpat neviens no minētajiem to nevar atņemt. Bet arī lietotājs, tāpat kā viduslaiku monarhs, uzņemas visas savu kļūdu sekas: nav regenta, kas pieņemtu lēmumus viņa vietā, ja viņš pazaudē zīmogu.

Izvēlei starp trešās puses pārvaldītu identitāti un pašsuverēnu identitāti nav vienas universāli pareizas atbildes. Sīka foruma kontam pārvaldīta identitāte, visticamāk, ir samērīga ar risku. Profesionālai identitātei, kas paraksta juridiski saistošus dokumentus, ekonomiskajai identitātei, kas sargā personīgos uzkrājumus, vai profesionālās saziņas identitātei ar klientiem, kuri uzticējuši sensitīvu informāciju, jautājums mainās. Tur jautājums vairs nav «vai tas ir ērti?», bet gan «kam, bez manis, ir vara rīkoties manu vārdā un kādos apstākļos?».

Kur šis mehānisms parādās reālās sistēmās

BIP39 radās Bitcoin pasaulē 2013. gadā un ātri izplatījās visā kriptovalūtu ekosistēmā: jebkurš nopietns maks šodien pieņem divpadsmit vai divdesmit četru vārdu BIP39 frāzi kā sava turētāja ekonomiskās identitātes

dublējumu. Ārpus kriptovalūtām tas pats pamatkoncepcijas elements — kriptogrāfiskais pāris, kas pierāda autorību bez starpnieka — parādās citās sistēmās ar atšķirīgu sintaksi. SSH atslēgas, kuras sistēmas administrators izmanto, lai piekļūtu saviem serveriem, ir klasisks piemērs: privātā atslēga, ko administrators glabā savā mašīnā, un publiskā, kas tiek nokopēta katrā serverī; neiejaucas neviena subjekta, kas būtu salīdzināma ar centralizētu pakalpojumu. Signal protokols izmanto Ed25519 ar pastāvīgu atslēgas materiālu ierīcē; Eiropas eIDAS savā kvalificētā paraksta daļā balstās uz to pašu kriptogrāfisko principu, tikai ar to atšķirību, ka atslēgu glabā kvalificēts uzticamības pakalpojumu sniedzējs, nevis lietotājs.

Solo2, šīs publikācijas izdevējplatforma, katra lietotāja identitātei izmanto divdesmit četru vārdu BIP39 frāzi. Lietotājs, izveidojot savu kontu, vārdus redz vienu reizi. Tie netiek glabāti nevienā Solo2 vai kāda cita serverī: ja lietotājs tos pieraksta un glabā, viņš saglabā savu identitāti uz visiem laikiem. Ja viņš tos pazaudē, viņš tos pazaudē. Tās ir loģiskas sekas arhitektūrai bez starpnieka operatora: ja Solo2 varētu atdot identitāti lietotājam, kurš to pazaudēja, tā varētu to atdot arī jebkuram, kurš izdarītu spiedienu uz Solo2, lai to saņemtu.

Profesionālam lasītājam

Četri apsvērumi tiem, kuri apsver iespēju pieņemt kriptogrāfisko pašsuverēno (autosoberana) identitāti profesionālā kontekstā:

1. Frāze ir identitāte. Fiziska glabāšana — papīrs, vairākas kopijas dažādās vietās, iespējams, iegravēts metāls ilgtermiņa lietošanai — piedāvā vairāk garantiju nekā digitālā glabāšana, kas palielina uzbrukuma virsmu, nesamazinot zaudējumu risku.
2. Atgūšana nav iespējama. Procesu izstrādāt, pieņemot, ka kādu dienu primārā kopija tiks pazaudēta, ir daudz prātīgāk nekā to atklāt zaudējuma dienā. Otra ģeogrāfiski nošķirta kopija atrisina gandrīz visus scenārijus.
3. Tas nav tas pats, kas eIDAS kvalificētais sertifikāts. Kvalificētam parakstam Savienībā — notariāliem aktiem, noteiktām procedūrām ar administrāciju — tiesību akti prasa kvalificētu sniedzēju, kurš glabā atslēgu. Kriptogrāfiskā pašsuverēnā identitāte kalpo profesionālai komunikācijai un dokumentu parakstīšanai ar pierādījumu vērtību, taču tā automātiski neaizstāj kvalificēto sertifikātu gadījumos, kad norma to prasa.
4. Ja identitāte ir jānodod — mantojums, profesionālā pēctecība, darbības izbeigšana — vēlams procedūru sagatavot pirms tam, nevis pēc tam. Formālas procedūras ar aploksnēm, kas aizzīmogotas ar zīmogvasku (lacre), norādījumi testamenta izpildītājam, deponēšana notāra birojā ir klasiski pasākumi, kas ir pilnībā saderīgi ar aktīva kriptogrāfisko raksturu.

Šis raksts noslēdz konceptuālo trio, kas uzsāka ciklu — hash, šifrēšana, identitāte —. Šīs trīs idejas balstās viena uz otru: hash sniedz nemainīgu nospiedumu, šifrēšana sniedz konfidencialitāti bez uzticamas trešās puses, identitāte sniedz autorību bez piešķirošas trešās puses. Visām trim ir kopīga īpašība, kas arī nav ideoloģiska: tās nodod no pakalpojuma pārvaldītāja lietotājam tehniskās iespējas, kas tradicionāli piederēja operatoram. Līdz ar tām tiek nodota arī atbildība. Godīga runāšana par jebkuru no šīm trim prasa runāt arī par pārējām divām.

Avoti un papildu literatūra

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, 2013. gada Bitcoin uzlabošanas priekšlikums. De facto standarts atkopšanas frāzēm kriptogrāfijas nozarē.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), ieskaitot Ed25519. IETF, 2017. gada janvāris. Normatīvā specifikācija parakstīšanas shēmai, ko izmanto lielā daļā mūsdienu nozares.
- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, 2.0 versija. IETF, 2000. gada septembris. Definē PBKDF2 algoritmu, ko izmanto BIP39 atvasināšanā no frāzes uz sēklu (seed).
- Regula (ES) 910/2014 (eIDAS) un tās attīstība ar Regulu (ES) 2024/1183 (eIDAS 2) — Eiropas ietvars elektroniskajai identitātei un kvalificētam parakstam. Režīms, kas atšķiras no pašsuverēnā, taču

konceptuāli balstīts uz tiem pašiem kriptogrāfiskajiem primitīviem.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Kanonisks teksts par pašsuverēnā modeļa principiem un saistībām, agrāks, taču būtisks, lai izprastu mūsdienu risinājumu saimi.

← IepriekšējaisBiznesa modelis kā uzticības signālsNākamais →Self-hosting kā profesionāla prakse

Jaunākie lasījumi

- Pārdomas · 2026. gada 29. jūnijs Tu neesi anonīms
- Pārdomas · 2026. gada 27. maijs Tas, ko paraksts nevar atrisināt
- Analīze · 2026. gada 26. maijs Patiesa vs šķietama konfidencialitāte: jautājumi, ko vērts sev uzdot

Paņemiet šo rakstu līdzi tur, kur jums nepieciešams.

↓ Markdown ↓ Parasts teksts ↓ PDF

Fails tiks lejupielādēts jūsu ierīcē. No turienes varat to saglabāt, importēt Solo2 vai kopīgot jebkur. Cuadernos nepieņem lēmumu par galamērķi jūsu vietā.

Vaska zīmogs · SHA-256 e5a251ffb7af7876b0a14b16a2b194df2b02567883cd7dfed520220b07e05102

Cuadernos Lacre · Menzuri Gestión S.L. publikācija ·
autors R.Eugenio · rediģējusi Solo2 komanda.