24 žodžiai: kas yra kriptografinė tapatybė

Kriptografinė tapatybė nėra slaptažodis: joks serveris jos saugo ir ji neatkuriama. Didaktinis BIP39 mechanizmo paaiškinimas, kodėl būtent dvidešimt keturi žodžiai ir kokia tikroji atsakomybė tenka tam, kuris juos valdo.

Kad suprastume vieni kitus: jei pamiršite „Gmail“ slaptažodį, „Google“ jį nustatys iš naujo. Jei prarasite 24 žodžius, sudarančius kriptografinę tapatybę, nėra ko jų prašyti. Ne todėl, kad procedūra būtų griežta – tiesiog kitame gale nieko nėra. Šis skirtumas yra esminis.

Skirtumas tarp slaptažodžio ir tapatybės

Slaptažodis klasikiniame interneto modelyje nėra vartotojo tapatybė. Tai įrodymas. Vartotojas turi tapatybę – vardą, el. paštą, kliento numerį – ir, norėdamas įrodyti serveriui, kad yra tas, kuo sakosi esąs, pateikia slaptažodį, kurį serveris palygina su išsaugotu atspaudu. Jei atspaudai sutampa, serveris leidžia prisijungti. Jei slaptažodis prarandamas, vartotojas lieka tuo pačiu vartotoju; tai, ką jis praranda, yra įrodymas, ir egzistuoja atkūrimo procedūra – el. laiškas registruotu adresu, saugos klausimas – jam susigrąžinti.

Kriptografinė tapatybė veikia kitaip. Tai nėra įgaliojimas, kurį kas nors lygina su išsaugotu atspaudu; tai *yra* pati savaime išbaigta matematinė paslaptis. Nesvarbu, kur ji saugoma – ant popieriaus, įrenginyje ar net svetimame serveryje: tapatybė egzistuoja dėl savo matematikos, o ne dėl to, kas ją patvirtina. Čia išryškėja savybė, panaši į tą, kurią matėme straipsnyje „Kas iš tikrųjų yra SHA-256“: valdymas įrodomas ne rodant paslaptį, o naudojant ją pasirašymui. Taip sukurtą parašą kiekvienas gali patikrinti naudodamas viešąją vertę, kuri matematiškai išvedama iš pačios paslapties, nebūtina žinoti pačios paslapties ir nereikia jokios trečiosios šalies tarpininkavimo. Kas turi paslaptį, tas ir yra tapatybė; kas ją praranda, tas nustoja ja būti. Verdiktas kategoriškas: **nėra nieko, ko būtų galima paprašyti grąžinti tapatybę. To kažko nėra, nes jis jos niekada ir neturėjo.**

Ką reprezentuoja dvidešimt keturi žodžiai

Kriptografinė tapatybė paprastai reprezentuojama trisdešimt dviejų baitų – dviejų šimtų penkiasdešimt šešių bitų – matematine paslaptimi. Skaičius, kurį sunku įsiminti ir dar sunkiau be klaidų nurašyti. Kriptografijos pramonė šią problemą išsprendė 2013 m. pasitelkusi nedidelį ir elegantišką standartą, vadinamą BIP39: būdą reprezentuoti tuos du šimtus penkiasdešimt šešis bitus kaip dvidešimt keturių žodžių seką, parinktą iš oficialaus dviejų tūkstančių keturiasdešimt aštuonių žodžių sąrašo. Po tuo slypinti aritmetika dera elegantiškai; kas nori ją pamatyti išsamiai, ras ją paraštėse.

Skaičiavimas prasideda nuo galo. Norime reprezentuoti du šimtus penkiasdešimt šešis paslapties bitus pridėdami aštuonis kontrolinės sumos bitus: iš viso du šimtus šešiasdešimt keturis bitus. Jei juos padalinsime į dvidešimt keturis žodžius – valdomą skaičių, kurį galima užrašyti ir padiktuoti be nuostolių – kiekvienas žodis turi suteikti tiksliai vienuolika informacijos bitų. O vienuolika bitų yra du pakelta vienuolintuoju laipsniu galimybių, t. y. du tūkstančiai keturiasdešimt aštuoni. Todėl oficialus BIP39 žodynas yra būtent tokio dydžio: sąrašas sudarytas pagal problemos mastą, o ne atvirkščiai.

Skaičiavimas nėra dekoratyvinis. Jei kas nors teisingai nurašys dvidešimt tris žodžius ir suklys rašydamas dvidešimt ketvirtąjį, kontrolinė suma tai aptiks: programinė įranga pasakys „ši seka negalioja“. Jei kas nors

teisingai nurašys visus dvidešimt keturis, programinė įranga vienareikšmiškai išves tą pačią tapatybę. Žodžių sąrašo pasirinkimas taip pat apgalvotas: BIP39 žodyno žodžiai yra trumpi, besiskiriantys vieni nuo kitų, be diakritinių ženklų, parinkti taip, kad būtų sumažinta fonetinė ir rašybos painiava. Tai žodynas, sukurtas tam, kad žmonės galėtų jį įsiminti, užrašyti ir padiktuoti be nuostolių.

Nuo frazės iki rakto

Tie dvidešimt keturi žodžiai nėra kriptografinis raktas, kuriuo pasirašomi pranešimai. Tai atkuriamas pradinės entropijos atvaizdas, kuris per deterministinį procesą, vadinamą PBKDF2, paverčiamas šešiasdešimt keturių baitų „sėkla“ (seed). Iš tos sėklos taip pat deterministiškai išvedami konkretūs kriptografiniai raktai, kuriuos naudoja vartotojas: privatus raktas pasirašymui ir atitinkamas viešasis raktas, kuris skelbiamas parašams tikrinti. Tas pats mechanizmas skirtingose sistemose: kriptovaliutos naudoja secp256k1 kreivę; Signal protokolas ir daugelis modernių sistemų naudoja Ed25519 Curve25519 kreivėje. Konkrečiai kreivei, pavyzdžiui, Ed25519, BIP32 ir SLIP-0010 standartai paima tą šešiasdešimt keturių baitų sėklą ir deterministiškai išveda trisdešimt du baitus, kurie sudaro efektyvų pasirašymo raktą — tuos pačius trisdešimt du baitus, kuriais prasideda kito skyriaus kodo pavyzdys.

Tai standartinis būdas, kuriuo visa pramonė pristato mechanizmą vartotojui —kriptovaliutų piniginės, decentralizuotos tapatybės valdytojai, Signal savo nuolatinės tapatybės dalyje, Solo2 tarp jų: vartotojas praktiškai niekada nemato sėklos ar išvestų raktų. Kurdamas savo tapatybę jis mato dvidešimt keturis žodžius ir pasirinktinai užsirašo juos ant popieriaus. Tada žodžiai keliauja tarp jo įrenginių, kai jis nori migruoti tapatybę: įveda juos į naują programėlę, programėlė išveda tą pačią sėklą, tuos pačius raktus, tą pačią tapatybę. Tai portabilus, kriptografiškai patikimas ir pagrįstose ribose įsimenamas mechanizmas.

Kaip pasirašyti raktu (Zig potėpis)

Zig kalboje, kai turite trisdešimt dviejų baitų sėklą, išvestą iš dvidešimt keturių žodžių, pranešimo pasirašymas su Ed25519 telpa į kelias eilutes:

```
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);
```

Pasirašymo operacija sukuria šešiasdešimt keturis baitus —vadinamus parašu— kuriuos buvo galima sugeneruoti tik iš atitinkamo privataus rakto. Patikra yra vieša: bet kas, turintis viešąjį raktą, gali patikrinti, ar parašas atitinka pranešimą. Be privataus rakto niekas negali sukurti galiojančio parašo tam pranešimui; su viešuoju raktu visi gali aptikti, ar parašas yra galiojantis. Ši asimetrija leidžia pasirašiusiajam įrodyti autorystę neatskleidžiant paslapties.

Ankstesnis pavyzdys yra minimali vadovo versija. Tikrajame Solo2 kode grandinė eina per du failus: vieną JavaScript kalba, kuris veikia vartotojo naršyklėje ir rekonstruoja entropiją iš dvidešimt keturių žodžių, kitą Zig kalba *zcatcrypto* bibliotekoje, kuris paima tą entropiją ir išveda konkrečius kriptografinius raktus. Pradedant nuo naršyklės pusės:

```
// 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 };
}
```

Tie trisdešimt du entropijos baitai kartu su kitais trisdešimt dviem, išvestais tame pačiame žingsnyje, keliauja į Zig WebAssembly modulį, kuris sugeneruoja pačius Ed25519 raktus. Pilna funkcija su galutiniu atminties išvalymu telpa viename ekrane:

```
// 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;
}
```

Verta pažymėti dvi detales. Pirma: ta pati pradinė reikšmė (seed) visada sukuria tą pačią raktų porą — būtent tai leidžia atkurti tapatybę įvedus dvidešimt keturis žodžius į naują įrenginį. Antra: paskutinėje eilutėje pradinė reikšmė yra aiškiai ištrinama iš atminties. Po šio taško net pati funkcija negalėtų rekonstruoti raktų; vienintelis šaltinis būtų vartotojo žodžiai.

Tiems, kurie nori tai patikrinti su mažais skaičiais. Pasirašymo schemą galima peržiūrėti visą su pakankamai mažais skaičiais, kad skaičiavimus būtų galima atlikti rankomis. Tie, kurie nenori gilintis į aritmetiką, gali praleisti šį bloką neprarasdami straipsnio esmės; tie, kurie nori pamatyti mechanizmą veikiantį žingsnis po žingsnio, ras jį čia. **Viešosios taisyklės**, kurias gali perskaityti bet kas: pirminis skaičius $p = 23$ (tikrajame Ed25519 jis yra apie septyniasdešimt septynių skaitmenų; naudojame dvidešimt tris, kad skaičiavimai tilptų viename puslapyje), pagrindas $g = 2$, kurio eilė šioje grupėje yra $q = 11$, ir konvencija, kad visa aritmetika su g atliekama *módulo p*, o visi eksponentai redukuojami *módulo q*. **Privatus pasirinkimas**, vienintelis ir niekada nesidalijamas: paslaptis $x = 6$. Tai ir yra tapatybė.

1 žingsnis — Viešoji tapatybės dalis. Ji apskaičiuojama vieną kartą ir viešai paskelbiama.

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

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

Viešoji tapatybės dalis yra **18**. Bet kas gali ją paimti ir naudoti parašams, sukurtiems su šia tapatybe, patikrinti. Niekas, stebėdamas tik skaičių 18, negali atkurti paslapties 6: tai diskretaus logaritmo problema, prie kurios grįšime pabaigoje.

2 žingsnis — Pranešimo pasirašymas. Tapatybės savininkas nori pasirašyti pranešimą $m = 7$. Jis pradeda pasirinkdamas naują atsitiktinę reikšmę $k = 4$, kuri bus naudojama tik vieną kartą ir niekada nebus dalijamasi (tikrajame Ed25519 k išvedamas deterministiškai iš pranešimo ir paslapties, kad būtų išvengta pakartotinio panaudojimo pavojaus, tačiau jo vaidmuo yra būtent toks). Tada jis apskaičiuoja tris skaičius:

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

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

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

Parašas yra pora **(r, s) = (16, 10)**. Jis keliauja atvirai kartu su pranešimu. Bet kas gali jį perskaityti. Didaktinė pastaba: tikrajame Ed25519 funkcija H yra SHA-512, kriptografiškai patikima; čia naudojame supaprastinimą $e = (r + m) \bmod q$, kad skaitytojas galėtų pereiti žingsnius be būtinybės skaičiuoti maišos reikšmę (hash). Algoritmo struktūra yra tokia pati.

3 žingsnis — Parašo tikrinimas. Tikrintojas turi viešąją dalį $y = 18$, pranešimą $m = 7$ ir parašą $(r, s) = (16, 10)$. Jis rekonstruoja e tuo pačiu būdu — $e = (16 + 7) \bmod 11 = 1$ — ir patikrina, ar ši lygybė galioja:

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

Apskaičiuoja abi puses atskirai:

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

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

Abi pusės duoda **12**. Parašas galioja. Bet kas, turintis viešąją dalį 18, gali padaryti šią išvadą niekada nežinojęs, kad paslaptis buvo 6.

O kaip trečioji šalis, kuri bandytų suklastoti? Eva matė viską viešai einant per kanalą: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Norėdama pasirašyti *kitokį* pranešimą šios tapatybės vardu, ji turėtų žinoti x. Vienintelis jos būdas yra paklausti savęs: „kokiam eksponentui x galioja $2^x \bmod 23 = 18$?“. Su $p = 23$ ji gali išbandyti 0, 1, 2, 3, … ir rasti jį per kelias sekundes. Tačiau pakeitus 23 pirminiu skaičiumi iš realiųjų Ed25519 matmenų, galimų eksponentų erdvė viršija atomų skaičių matomoje visatoje. **Šiandien žmonijai nėra žinomas joks algoritmas, kuris galėtų pereiti tą erdvę per mažiau nei milijardus metų.** Tai ta pati diskretaus logaritmo problema, kuri yra ankstesnio straipsnio Diffie-Hellman pagrindas, čia pritaikyta pasirašymo schemai.

Tai, ką ką tik perėjome, yra *tiksliai* Schnorr, pasirašymo schema, kurios variantas yra elipsinei kreivei pritaikytas Ed25519. Tikrajame Ed25519 visos operacijos atliekamos su konkrečios kreivės (Curve25519) taškais, o ne su sveikaisiais skaičiais modulo pirminis skaičius, o funkcija H yra SHA-512 vietoj aukščiau naudotos paprastos sumos. Abu pakeitimai yra diegimo patobulinimai — kriptografinio atsparumo grubiai jėgai įgijimas, papildomų saugumo savybių įgijimas k. Algoritminė struktūra, trys operacijos ir asimetrijos priežastis yra tos pačios.

Čia pravartu trumpam sustoti, nes visa grandinė iš pirmo žvilgsnio gali būti supainiota su kitu trejeto primityvu: maiša (hash). Tai ne ji. Maiša yra unikali funkcija, kuri suspaudžia — įeina daug baitų, išeina trumpas atspaudas, tuo kelias baigiasi. Kriptografinė tapatybė yra matematiškai papildanti viena kitą pora: paslaptis lieka ir pasirašo; jos viešoji dalis paskelbiama ir patikrina. Ten, kur maiša sutraukia informaciją viena kryptimi, tapatybė nustato asimetriją tarp dviejų pusių. Maiša patvirtina, kas buvo pasakyta; tapatybė patvirtina, kas tai pasakė.

Kuo frazė nėra

Reikėtų išsklaidyti tris dažnus klaidingus įsitikinimus. Frazė nėra slaptažodis tikrąja prasme: ji nėra lyginama su serveryje saugomu piršto atspaudu; ji įvedama į vartotojo įrenginį, kad matematiškai atkurtų tapatybę. Frazė nėra atkuriama: jei ji pamesta, nėra ko paprašyti ją grąžinti; jei ji nukopijuojama, nukopijuojama ir tapatybė. Frazė nėra nuo tapatybės atskiriamas kredencialas: frazė *yra* tapatybė. Kas ją turi, gali veikti kaip ta tapatybė be papildomo leidimo, be autorizacijos proceso, be galimybės atkurti.

Būtent ši trečioji savybė keičia reikalų svorį. Pamestas slaptažodis yra administracinis nepatogumas. Pamesta kriptografinė tapatybė yra tapatybės praradimas. Trečiųjų šalių rastas popierius su fraze nėra paskyros vagystės rizika: tai visos tapatybės perdavimas. Sistemos pažadas — kad niekas negali panaikinti jūsų tapatybės ar savavališkai jūsų užblokuoti — yra neatsiejamai susijęs su atsakomybe — kad jūs esate vienintelis saugotojas to, ko niekas negali jums sugrąžinti.

Pažadas ir svoris

Kriptografinės tapatybės modelis dažnai apibūdinamas kaip *savivaldis* (angl. self-sovereign). Žodžio pasirinkimas yra apgalvotas ir gana tiksliai apibūdina būseną. Vartotojas yra savo tapatybės suverenas beveik viduramžiška prasme: jos nesuteikia joks karalius, joks išdavėjas ar jokia centrinė valdžia; taip pat niekas iš išvardytų negali jos atimti. Bet, kaip ir viduramžių monarchas, vartotojas neša visą savo klaidų atsakomybę: nėra regento, kuris priimtų sprendimus už jį, jei jis pamestų antspaudą.

Pasirinkimas tarp tapatybės, kurią valdo trečioji šalis, ir savivaldės tapatybės neturi vieno universalaus teisingo atsakymo. Nesvarbios forumo paskyros atveju valdoma tapatybė tikriausiai yra proporcinga rizikai. Profesinės tapatybės, pasirašančios teisiškai įpareigojančius dokumentus, ekonominės tapatybės, saugančios asmenines santaupas, ar profesinės komunikacijos tapatybės su klientais, kurie patikėjo jautrią informaciją, atveju reikalas pasikeičia. Ten klausimas nustoja būti „ar tai patogu?“ ir tampa „kas, be manęs, turi galią veikti kaip aš ir kokiomis aplinkybėmis?“.

Kur šis mechanizmas pasirodo realiose sistemose

BIP39 gimė Bitcoin pasaulyje 2013 m. ir greitai išplito visoje kriptovaliutų ekosistemoje: bet kuri rimta piniginė šiandien priima dvylikos arba dvidešimt keturių žodžių BIP39 frazę kaip savo turėtojo ekonominės tapatybės atsarginę kopiją. Be kriptovaliutų, ta pati pagrindinė koncepcija — kriptografinė pora, įrodanti autorystę be tarpininko — pasirodo kitose sistemose su skirtinga sintakse. SSH raktai, kuriuos sistemų administratorius naudoja prieigai prie savo serverių, yra klasikinis atvejis: privatus raktas, kurį administratorius saugo savo mašinoje, ir viešasis, kuris nukopijuojamas į kiekvieną serverį; joks subjektas, palyginamas su centralizuota paslauga, nesikiša. Signal protokolas naudoja Ed25519 su nuolatine rakto medžiaga įrenginyje; Europos eIDAS savo kvalifikuoto parašo dalyje remiasi tuo pačiu kriptografiniu principu, tik skirtumas tas, kad raktą saugo kvalifikuotas patikimumo paslaugų teikėjas, o ne vartotojas.

Solo2, šio leidinio leidybos platforma, kiekvieno vartotojo tapatybei naudoja dvidešimt keturių žodžių BIP39 frazę. Vartotojas, kurdamas savo paskyrą, žodžius pamato vieną kartą. Jie nesaugomi jokiame Solo2 ar kieno nors kito serveryje: jei vartotojas juos užsirašo ir saugo, jis išlaiko savo tapatybę visam laikui. Jei juos pameta, pameta. Tai yra nuosekli architektūros be tarpininko operatoriaus pasekmė: jei Solo2 galėtų grąžinti tapatybę vartotojui, kuris ją pametė, ji taip pat galėtų ją atiduoti bet kam, kas darytų spaudimą Solo2, kad ją gautų.

Profesionaliam skaitytojui

Keturi aspektai tiems, kurie svarsto galimybę priimti kriptografinę savarankišką (autosoberana) tapatybę profesiniame kontekste:

1. Frazė yra tapatybė. Fizinis saugojimas — popierius, kelios kopijos skirtingose vietose, galiausiai išgraviruotas metalas ilgalaikiam naudojimui — suteikia daugiau garantijų nei skaitmeninis saugojimas, kuris padidina atakos paviršių nesumažindamas praradimo rizikos.
2. Atkūrimo nėra. Suprojektuoti procesą darant prielaidą, kad vieną dieną pirminė kopija bus prarasta, yra daug išmintingiau nei tai sužinoti praradimo dieną. Antra geografiškai atskirta kopija išsprendžia beveš visus scenarijus.
3. Tai nėra tas pats, kas eIDAS kvalifikuotas sertifikatas. Kvalifikuotam parašui Sąjungoje — notariniams aktams, tam tikroms procedūroms su administracija — teisės aktai reikalauja kvalifikuoto teikėjo, kuris saugo raktą. Kriptografinė savarankiška tapatybė pasitarnauja profesinei komunikacijai ir dokumentų pasirašymui, turinčiam įrodomąją galią, tačiau ji automatiškai nepakeičia kvalifikuoto sertifikato tais atvejais, kai norma to reikalauja.
4. Jei tapatybė bus perduodama — palikimas, profesinis perėmimas, veiklos nutraukimas — patartina procedūrą parengti prieš tai, o ne po to. Formalios procedūros su antspauduotu vašku (lacre) užklijuotais vokais, instrukcijos testamento vykdytojui, deponavimas notaro biure yra klasikiniai susitarimai, puikiai suderinami su kriptografine turto prigimtimi.

Šis straipsnis užbaigia koncepcinę trijulę, pradėjusią ciklą — hash, šifravimas, tapatybė —. Šios trys idėjos remiasi viena kita: hash suteikia nekintantį atspaudą, šifravimas suteikia konfidencialumą be patikimos trečiosios šalies, tapatybė suteikia autorystę be leidimą suteikiančios trečiosios šalies. Visos trys turi savybę, kuri taip pat nėra ideologinė: jos perduoda iš paslaugą valdančiojo vartotojui technines galimybes, kurios tradiciškai priklausė operatoriui. Kartu su jomis perduodama ir atsakomybė. Sąžiningas kalbėjimas apie bet kurią iš šių trijų reikalauja kalbėti ir apie kitas dvi.

Šaltiniai ir papildomas skaitymas

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, 2013 m. Bitcoin tobulinimo pasiūlymas. De facto standartas atkūrimo frazėms kriptografijos pramonėje.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), įskaitant Ed25519. IETF, 2017 m. sausis. Normatyvinė pasirašymo schemos, naudojamos didelėje dalyje šiuolaikinės pramonės, specifikacija.

- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, 2.0 versija. IETF, 2000 m. rugsėjis. Apibrėžia PBKDF2 algoritmą, naudojamą BIP39 išvedimui iš frazės į sėklą (seed).
- Reglamentas (ES) 910/2014 (eIDAS) ir jo raida Reglamentu (ES) 2024/1183 (eIDAS 2) — Europos elektroninės tapatybės ir kvalifikuoto parašo sistema. Režimas, besiskiriantis nuo savarankiško, tačiau konceptualiai pagrįstas tais pačiais kriptografiniais primityvais.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Kanoninis tekstas apie savarankiško modelio principus ir įsipareigojimus, ankstesnis, tačiau svarbus norint suprasti šiuolaikinių sprendimų šeimą.

← AnkstesnisVerslo modelis kaip pasitikėjimo signalasKitas →Self-hosting kaip profesinė praktika

Naujausi skaitiniai

- Apmąstymas · 2026 m. birželio 29 d. Jūs nesate anonimiškas
- Refleksija · 2026 m. gegužės 27 d. Ko parašas negali ištaisyti
- Analizė · 2026 m. gegužės 26 d. Tikras vs tariamas privatumas: klausimai, kuriuos verta užduoti sau

Pasiimkite šį straipsnį su savimi ten, kur jums reikia.

↓ Markdown ↓ Paprastas tekstas ↓ PDF

Failas bus atsisiųstas į jūsų įrenginį. Iš ten galite jį išsaugoti, importuoti į Solo2 arba bendrinti bet kur. Cuadernos nusprendžia ne jūsų naudai dėl paskirties vietos.

Vaško antspaudas · SHA-256 a59650b9f0ce37ba93d4c13f118cf88e14420cb570f8322d903f002b5773294a

Cuadernos Lacre · Menzuri Gestión S.L. leidinys ·
parašė R.Eugenio · redagavo Solo2 komanda.