Kontzeptua · 2026(e)ko maiatzaren 23(a)

← Aurrekoa·Hurrengoa →

24 hitzak: zer den identitate kriptografiko bat

Identitate kriptografiko bat ez da pasahitz bat: zerbitzaririk ez du gordetzen eta ez da berreskuratzen. BIP39 mekanismoaren azalpen didaktiko bat, zergatik zehazki hogeita lau hitz, eta zer pisu erreal jausten den haiek dituenaren gain.

Elkar ulertzeko: Gmail-eko pasahitza ahazten baduzu, Google-k berrezarri egiten dizu. Identitate kriptografiko bat osatzen duten 24 hitzak galtzen badituzu, ez dago nori eskatu. Ez da prozedura zorrotza dela — kontua da ez dagoela inor beste aldean. Diferentzia hori da diferentzia guztia.

Pasahitz baten eta identitate baten arteko aldea

Pasahitz bat, interneteko eredu klasikoan, ez da erabiltzailearen identitatea. Egiaztagiri bat da. Erabiltzaileak identitate bat du —izen bat, helbide elektroniko bat, bezero-zenbaki bat— eta, zerbitzari baten aurrean nor den frogatzeko, zerbitzariak gorde zuen aztarna batekin alderatzen duen pasahitz bat aurkezten du. Aztarnak bat badatoz, zerbitzariak saioa ematen du. Pasahitza galtzen bada, erabiltzaileak erabiltzaile bera izaten jarreatzen du; galtzen duena egiaztagiria da, eta berreskuratzeko prozedura bat dago —erregistratutako helbidera mezu bat, segurtasun-galdera bat— hura leheneratzeko.

Identitate kriptografiko batek beste modu batera funtzionatzen du. Ez da norbaitek aztarna baten aurrean alderatzen duen kredentzial bat; *berez* sekretu matematiko oso bat da. Berdin dio non dagoen —paper batean, gailu batean, baita besteren zerbitzari batean ere—: identitatea bere matematikagatik existitzen da, ez balidatzen duenagatik. Hemen «Zer den benetan SHA-256» artikuluan ikusi genuenaren antzeko propietate bat agertzen da: jabetza ez da sekretua erakutsiz frogatzen, sinatzeko erabiliz baizik. Horrela sortutako sinadura edonork egiaztatu dezake sekretutik beretik matematikoki eratortzen den balio publiko batekin, sekretua bera ezagutu beharrik gabe, eta hirugarren baten bitartekaritzarik gabe. Sekretua duena, identitatea da; galtzen duena, izateari uzten dio. Sententzia kategorikoa da: **ez dago nori eskatu identitatea itzultzeko. Inor hori ez da existitzen, ez zuelako hasieratik ere.**

Hogeita lau hitzek ordezkatzen dutena

Identitate kriptografikoa hogeita hamabi byteko —berrehun eta berrogeita hamasei biteko— sekretu matematiko baten bidez irudikatu ohi da. Gogoratzeko zaila den eta akatsik gabe transkribatzeko are zailagoa den zenbaki bat. Kriptografia-industriak 2013an konpondu zuen arazo hori BIP39 izeneko estandar txiki eta dotore baten bidez: berrehun eta berrogeita hamasei bit horiek bi mila eta berrogeita zortzi hitzeko zerrenda ofizial batetik hartutako hogeita lau hitzeko sekuentzia gisa irudikatzeko modu bat. Atzean dagoen aritmetika dotoreziaz egokitzen da; xehetasunez ikusi nahi duenak marjinean aurkituko du.

Kontaketa amaieratik hasten da. Sekretuaren berrehun eta berrogeita hamasei bitak irudikatu nahi ditugu, zortzi bit-eko kontrol-batura (checksum) gehituz: berrehun eta hirurogeita lau bit guztira. Hogeita lau hitzetan banatzen baditugu —galerarik gabe idazteko eta diktatzeko moduko kopurua—, hitz bakoitzak hamaika bit informazio eman behar ditu zehazki. Eta hamaika bit bi ber hamaika aukera dira, hau da, bi mila eta berrogeita zortzi. Horregatik du BIP39 hiztegi ofizialak tamaina hori zehazki: zerrenda arazoaren neurrira existitzen da, eta ez alderantziz.

Kontaketa ez da apaingarria. Norbaitek hogeita hiru hitz ondo transkribatzen baditu eta hogeita laugarrenean hanka sartzen badu, kontrol-baturak hauteman egingo du: softwareak «sekuentzia hau ez da baliozkoa» esango dio. Hogeita laurak ondo transkribatzen baditu, softwareak identitate bera eratorriko du anbiguotasunik gabe. Hitz-zerrendaren aukeraketa ere nahita egindakoa da: BIP39 hiztegiko hitzak laburrak dira, elkarren artean desberdinak, diakritikorik gabeak, nahasmendu fonetiko eta ortografikoak minimizatzeko aukeratuak. Gizakiek galerarik gabe gogoratzeko, idazteko eta diktatzeko diseinatutako hiztegia da.

Esaldi batetik gakora

Hogei eta lau hitzak ez dira mezuak sinatzen dituen gako kriptografikoa. Jatorrizko entropiaren irudikapen berreskuragarri bat dira, eta PBKDF2 izeneko prozesu determinista baten bidez, hirurogeita lau byteko hazi (seed) bihurtzen dira. Hazi horretatik eratortzen dira, baita modu deterministan ere, erabiltzaileak erabiltzen dituen gako kriptografiko zehatzak: sinatzeko gako pribatu bat eta sinadurak egiaztatzeko argitaratzen den gako publiko bat. Mekanismo bera sistema desberdinetan: kriptomonetek secp256k1 kurba erabiltzen dute; Signal protokoloak eta sistema moderno askok Ed25519 erabiltzen dute Curve25519 kurbaren gainean. Ed25519 bezalako kurba zehatz baterako, BIP32 eta SLIP-0010 estandarrek hirurogeita lau byteko hazi hori hartzen dute eta, modu deterministan, sinadura-gako eraginkorra osatzen duten hogeita hamabi byteak eratortzen dituzte —hurrengo ataleko kode-adibidea hasten den hogeita hamabi byte berberak.

Hau da industria osoak mekanismoa erabiltzaileari aurkezteko modu estandarra —kriptomoneta-zorroak, identitate-kudeatzaile deszentralizatuak, Signal bere identitate iraunkorraren zatian, Solo2 horien artean—: erabiltzaileak, praktikan, ez du inoiz hazia edo eratorritako gakoak ikusten. Hogei eta lau hitzak ikusten ditu bere identitatea sortzean eta, nahi izanez gero, paper batean idazten ditu. Ondoren, hitzak bere gailuen artean bidaiatzen dute identitatea migratu nahi duenean: aplikazio berrian sartzen ditu, aplikazioak hazi bera eratortzen du, gako berberak, identitate bera. Mekanismo eramangarria da, kriptografikoki sendoa eta, arrazoizko mugen barruan, gogoragarria.

Nola sinatu gakoarekin (Zig pintzelkada)

Zig-en, hogei eta lau hitzetatik eratorritako hogeita hamabi byteko hazia daukazunean, mezu bat Ed25519-rekin sinatzea lerro gutxitan sartzen da:

```
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);
```

Sinatze-eragiketak hirurogeita lau byte sortzen ditu —sinadura izenekoak—, dagokion gako pribatutik soilik sor litezkeenak. Egiaztapena publikoa da: gako publikoa duen edonork egiaztatu dezake sinadura mezuari dagokiola. Gako pribaturik gabe, inork ezin du mezu horretarako baliozko sinadurarik sortu; gako publikoarekin, denek detektatu dezakete sinadura bat baliozkoa den. Asimetria horri esker, sinatzaileak egiletza frogatu dezake sekretua partekatu gabe.

Aurreko adibidea eskuliburuaren bertsio minimoa da. Solo2-ren benetako kodean, kateak bi fitxategi zeharkatzen ditu: bata JavaScript-en, erabiltzailearen nabigatzailean bizi dena eta hogeita lau hitzetatik entropia

berreraikitzen duena; bestea Zig-en, *zcatcrypto* liburutegiaren barruan, entropia hori hartu eta gako kriptografiko zehatzak eratortzen dituena. Nabigatzailearen aldetik hasita:

```
// 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 };
}
```

Entropiaren hogeita hamabi byte horiek, urrats berean eratorritako beste hogeita hamabirekin batera, Zig-en WebAssembly modulura bidaiatzen dute, benetako Ed25519 gakoak sortzen dituena. Funtzio osoa, bere azken memoria-garbiketa eta guzti, pantaila bakarrean kabitzen da:

```
// 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;
}
```

Bi xehetasun azpimarratzea merezi du. Lehenengoa: hazi berak beti gako-pare bera sortzen du —horrek ahalbidetzen du, hain zuzen, identitatea berreskuratzea hogeita lau hitzak gailu berri batean sartuz. Bigarrena: hazia esplizituki ezabatzen da memoriatik azken lerroan. Puntu horretatik aurrera, funtzioak berak ere ezingo lituzke gakoak berreraiki; erabiltzailearen hitzak lirateke jatorri bakarra.

Zenbaki txikiekin egiaztatu nahi duenarentzat. Sinadura-eskema oso-osorik zeharka daiteke kalkuluak eskuz egiteko bezain zifra txikiekin. Aritmetikan sartu nahi ez duenak bloke hau saltatu dezake artikuluaren haria galdu gabe; mekanismoa pausoz pauso nola dabilen ikusi nahi duenak hemen aurkituko du. **Arau publikoak**, edonork irakur ditzakeenak: $p = 23$ zenbaki lehena (benetako Ed25519-an hirurogeita hamazazpi bat zifratakoa da; hogeita hiru erabiltzen ditugu kalkuluak orrialde batean kabitzeko), $g = 2$ oinarria, talde honetan bere ordena $q = 11$ duena, eta *g*-rekiko aritmetika guztia *módulo p* egiten dela eta berretzaile guztiak *módulo q* murrizten direla dioen hitzarmena. **Aukera pribatua**, bakarra eta inoiz partekatzen ez dena: $x = 6$ sekretua. Hori da identitatea.

1. urratsa — Identitatearen zati publikoa. Behin kalkulatzen da eta modu irekian argitaratzen da.

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

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

Identitatearen zati publikoa **18** da. Edonork hartu eta erabil dezake identitate honekin egindako sinadurak egiaztatzeko. Inork ezin du $x = 6$ sekretua berreskuratu 18a soilik behatuz: hori da logaritmo diskretuaren arazoa, amaieran berriro aztertuko duguna.

2. urratsa — Mezu bat sinatzea. Identitatearen jabeak $m = 7$ mezua sinatu nahi du. Alde batetik $k = 4$ ausazko balio berri bat aukeratzen du; behin bakarrik erabiliko da eta ez da inoiz partekatuko (benetako Ed25519-an, *k* mezutik eta sekretutik eratorria da modu deterministan, berriro erabiltzeko arriskua saihesteko, baina betetzen duen papera hau da hain zuzen). Ondoren, hiru zenbaki kalkulatzen ditu:

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

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

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

Sinadura **(r, s) = (16, 10)** parea da. Mezuekin batera modu irekian bidaiatzen du. Edonork irakur dezake. Ohar didaktikoa: benetako Ed25519-an *H* funtzioa SHA-512 da, kriptografikoki sendoa; hemen $e = (r + m) \bmod q$ sinplifikazioa erabiltzen dugu irakurleak urratsak zeharka ditzan hasha kalkulatu beharrik gabe. Algoritmoaren egitura bera da.

3. urratsa — Sinadura egiaztatzea. Egiaztatzaileak $y = 18$ zati publikoa, $m = 7$ mezua eta $(r, s) = (16, 10)$ sinadura ditu. *e* berreraikitzen du modu berean —$e = (16 + 7) \bmod 11 = 1$— eta berdintza hau betetzen den egiaztatzen du:

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

Bi aldeak bereizita kalkulatzen ditu:

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

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

Bi aldeek **12** ematen dute. Sinadura baliozkoa da. 18 zati publikoa duen edonor irits daiteke ondorio honetara sekretua 6 zela jakin gabe.

Eta faltsutzen saiatzen den hirugarren bat? Evak kanaletik igarotzen den guztia ikusi du: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Identitate honen izenean *bestelako* mezu bat sinatzeko, x ezagutu beharko luke. Bere bide bakarra galdera hau egitea da: «zein x berretzailerako betetzen da $2^x \bmod 23 = 18$?». $p = 23$ izanda, 0, 1, 2, 3, … proba ditzake eta segundotan aurkitu. Baina 23aren ordez benetako Ed25519-aren dimentsioetako lehen bat jartzean, berretzaile posibleen espazioak unibertso behagarriko atomo kopurua gainditzen du. **Gaur egun ez dago gizateriak ezagutzen duen algoritmorik espazio hori milaka milioi urte baino gutxiagoan zeharkatu dezakeenik.** Aurreko artikuluko Diffie-Hellman oinarritzen duen logaritmo diskretuaren arazo bera da, hemen sinadura-eskemari aplikatua.

Zeharkatu berri duguna *zehazki* Schnorr da, Ed25519 bera kurba eliptiko batera egokitutako aldaera bat den sinadura-eskema. Benetako Ed25519-an, eragiketa guztiak kurba zehatz baten puntuen gainean egiten dira (Curve25519), zenbaki osoen gainean egin beharrean, eta H funtzioa SHA-512 da, goian erabili dugun jostailuzko batuketa izan beharrean. Bi ordezkapenak inplementazio-doikuntzak dira —indar gordinaren aurkako erresistentzia kriptografikoa irabaztea, k-rako segurtasun-propietate gehigarriak irabaztea—. Egitura algoritmikoa, hiru eragiketak eta asimetriaren arrazoia berdinak dira.

Hemen etenaldi labur bat egitea komeni da, kate osoa hirukotearen beste primitiba batekin nahas baitaiteke gainetik begiratuta: hasha. Ez da hori. Hasha konprimatzen duen funtzio bakarra da —byte asko sartzen dira, aztarna labur bat ateratzen da, eta hor amaitzen da bidea—. Identitate kriptografikoa bikote matematiko osagarria da: sekretua geratu eta sinatu egiten du; bere kide publikoa argitaratu eta egiaztatu egiten da. Hashak informazioa norabide bakarrean kolapsatzen duen lekuan, identitateak asimetria bat ezartzen du bi erdien artean. Hashak zer esan zen ziurtatzen du; identitateak nork esan zuen ziurtatzen du.

Esaldiak ez dena

Oso ohikoak diren hiru oker argitzea komeni da. Esaldia ez da pasahitz bat zentzu hertsian: ez da zerbitzari batean gordetako hatz-marka batekin konparatzen; erabiltzailearen gailuan sartzen da identitatea matematikoki berreraikitzeko. Esaldia ez da berreskuratzen: galtzen bada, ez dago inori eskatzeko; bikoizten bada, identitatea ere bikoizten da. Esaldia ez da identitatetik bereiz daitekeen kredentzial bat: esaldia identitatea *da*. Hura duenak identitate gisa joka dezake, baimen gehigarririk gabe, baimen-prozesurik gabe, berreskuratzeko aukerarik gabe.

Hirugarren propietate hori da kontuaren pisua aldatzen duena. Galdutako pasahitza eragozpen administratiboa da. Galdutako identitate kriptografikoa identitatea bera da. Hirugarrenek aurkitutako esaldia duen paper bat ez da kontua lapurtzeko arriskua: identitate osoa entregatzea da. Sistemaren agindua —inork ezin dizula zure identitatea kendu edo arbitrarioki blokeatu— erantzukizunarekin batera dator bereiztezin: inork zure ordez berreskuratu ezin duen zerbaiten zaindari bakarra zarela.

Promesa eta pisua

Identitate kriptografikoaren ereduari *identitate burujabea* deitu ohi zaio —self-sovereign ingelesezko literaturan—. Hitzaren aukeraketa nahitakoa da eta egoera nahiko zehatz deskribatzen du. Erabiltzailea bere identitatearen jabe da zentzu ia Erdi Arokoan: ez du inolako erregek, jaulkitzailek edo agintaritza zentralek ematen; aurrekoetako inork ezin du kendu ere. Baina, Erdi Aroko monarkak bezala, erabiltzaileak bere akatsen ondorio osoa bere gain hartzen du: ez dago bere ordez erabakiak hartuko dituen erregeorderik zigilua galtzen badu.

Hirugarren batek kudeatutako identitatearen eta identitate burujabearen arteko aukeraketak ez du erantzun unibertsal zuzenik. Garrantzirik gabeko foro bateko kontu baterako, kudeatutako identitatea arriskuarekiko proportzionala izango da seguruenik. Legez lotesleak diren dokumentuak sinatzen dituen identitate profesional baterako, aurrezkiak zaintzen dituen identitate ekonomiko baterako, informazio sentikorra konfiantzaz eman duten bezeroekin komunikazio profesionala duen identitate baterako, kontua aldatu egiten da. Hor, galdera ez da

«erosoa al da?» eta honako hau bihurtzen da: «nork dauka, nitaz gain, nire izenean jokatzeko boterea, eta zein egoeratan?».

Non agertzen den mekanismo hau benetako sistemetan

BIP39 Bitcoin munduan jaio zen 2013an, eta azkar hedatu zen kriptomoneten ekosistema osora: edozein zorro seriok onartzen du gaur egun hamabi edo hogeita lau hitzeko BIP39 esaldi bat bere jabearen nortasun ekonomikoaren babeskopia gisa. Kriptomonetetatik kanpo, azpiko kontzeptu bera —bitartekaririk gabeko egiletza frogatzen duen bikote kriptografikoa— beste sistema batzuetan agertzen da, sintaxi ezberdinarekin. Sistema-administratzaile batek bere zerbitzarietara sartzeko erabiltzen dituen SSH gakoak kasu klasiko bat dira: administratzaileak bere makinan gordetzen duen gako pribatu bat eta zerbitzari bakoitzean kopiatzen den gako publiko bat; ez du zerbitzu zentralizatu baten pareko inongo erakundek esku hartzen. Signal protokoloak Ed25519 erabiltzen du gailuan gako-material iraunkorrarekin; Europako eIDASek, sinadura kualifikatuaren zatian, printzipio kriptografiko berean oinarritzen dira, gakoa erabiltzailearen ordez konfiantzazko zerbitzu-hornitzaile kualifikatu batek zaintzen duelako aldearekin.

Solo2k, argitalpen honen argitaletxe-plataformak, hogeita lau hitzeko BIP39 esaldi bat erabiltzen du erabiltzaile bakoitzaren nortasun gisa. Erabiltzaileak, bere kontua sortzean, behin ikusten ditu hitzak. Ez dira Solo2ko inongo zerbitzaritan gorde, ezta beste inoren zerbitzaritan ere: erabiltzaileak hitzak idatzi eta zaintzen baditu, bere nortasuna mantentzen du betiko. Galtzen baditu, galdu egin ditu. Tartean operatzailerik ez dagoen arkitekturaren ondorio koherentea da: Solo2k nortasuna galdu duen erabiltzaileari nortasuna itzuli ahal balio, Solo2 presionatzen duenari ere eman ahal izango lioke.

Irakurle profesionalarentzat

Lau gogoeta testuinguru profesional batean nortasun kriptografiko autosoberanoa (autosoberana) hartzea ebaluatzen duenarentzat:

1. Esaldia nortasuna da. Zainketa fisikoak —papera, kopia batzuk leku ezberdinetan, azkenik epe luzerako grabatutako metala— berme gehiago eskaintzen ditu zainketa digitalak baino; izan ere, zainketa digitalak Eraso-azalera gehitzen du, galera-arriskua murriztu gabe.
2. Ez dago berreskurapenik. Prozesua egunen batean kopia primarioa galduko dela onartuz diseinatzea askoz egokiagoa da galdu den egunean horretaz jabetzea baino. Geografikoki banatutako bigarren kopia batek eszenatoki ia guztiak konpontzen ditu.
3. Ez da eIDAS ziurtagiri kualifikatu baten gauza bera. Batasunean sinadura kualifikatua lortzeko —notario-eskriturak, Administrazioarekin egin beharreko izapide batzuk—, legediak gakoa zaintzen duen hornitzaile kualifikatu bat eskatzen du. Nortasun kriptografiko autosoberanoak komunikazio profesionalerako eta froga-balioa duen dokumentu-sinadurarako balio du, baina ez du automatikoki ziurtagiri kualifikatua ordezkatzen arauak hala eskatzen duen kasuetan.
4. Nortasuna transferitu behar bada —herentzia, ondorengotza profesionala, jarduera-etetea—, komeni da prozedura lehenago prestatzea, ez geroago. Lacre-zigiluarekin (lacre) itxitako gutun-azalak dituzten prozedura formalak, testamentu-betearazle bati emandako jarraibideak, notarioan gordailutzea, aktiboaren izaera kriptografikoarekin guztiz bateragarriak diren konponbide klasikoak dira.

Artikulu honek zikloa ireki zuen hirukote kontzeptuala —hash, zifratzea, nortasuna— ixten du. Hiru ideiak elkarren gainean eraikitzen dira: hashak aztarna aldaezina ematen du, zifratzeak konfidentzialtasuna ematen du konfiantzazko hirugarrenik gabe, nortasunak egiletza ematen du hirugarren emalerririk gabe. Hirurek ideologikoa ez den jabetza bat partekatzen dute: zerbitzu bat kudeatzen duenarengandik erabiltzen duenarengana eramaten dituzte tradizioz operadorearengan zeuden gaitasun teknikoak. Horiekin batera, erantzukizunak ere eramaten dituzte. Hiruetako edozeini buruz zintzotasunez hitz egiteak beste biei buruz ere hitz egitea eskatzen du.

Iturriak eta irakurgai gehiago

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, 2013ko Bitcoin hobetzeko proposamena. Kriptoindustrian berreskuratze-esaldietarako de facto estandarra.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), Ed25519 barne. IETF, 2017ko urtarrila. Gaur egungo industria-zati handi batean erabiltzen den sinadura-eskemaren zehaztapen normatiboa.
- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, 2.0 bertsioa. IETF, 2000ko iraila. Esalditik hazira (seed) deribatzeko BIP39an erabiltzen den PBKDF2 algoritmoa definitzen du.
- 910/2014 (eIDAS) Erregelamendua (EB) eta 2024/1183 (eIDAS 2) Erregelamenduak (EB) emandako bilakaera — nortasun elektronikorako eta sinadura kualifikaturako Europako esparrua. Autosoberanoaren bestelako erregimena, baina kontzeptualki oinarri kriptografiko berberetan oinarritua.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Eredu autosoberanoaren printzipioei eta konpromisoei buruzko testu kanonikoa, lehenagokoa baina gaur egungo soluzioen familia ulertzeko garrantzitsua.

← AurrekoaNegozio-eredua konfiantza-seinale gisaHurrengoa →Self-hosting praktika profesional gisa

Azken irakurketak

- Gogoeta · 2026(e)ko ekainaren 29(a) Ez zara anonimoa
- Hausnarketa · 2026(e)ko maiatzaren 27(a) Sinadura batek konpondu ezin duena
- Analisia · 2026(e)ko maiatzaren 26(a) Pribatutasun erreala vs itxurazkoa: geure buruari egin beharreko galderak

Eraman artikulu hau zurekin behar duzun lekura.

↓ Markdown ↓ Testu arrunta ↓ PDF

Fitxategia zure gailura deskargatuko da. Bertatik gorde, Solo2ra inportatu edo nahi duzun lekuan parteka dezakezu. Cuadernosek ez du helburua zure ordez erabakitzen.

Lakre-zigilua · SHA-256 854cea16144f4f473abbd3b41352d286ba38fbca7155fb4d7141bb8346163266

Cuadernos Lacre · Menzuri Gestión S.L.ren argitalpena ·
R.Eugeniok idatzia · Solo2 taldeak editatua.