24 sõna: mis on krüptograafiline identiteet

Krüptograafiline identiteet ei ole parool: ükski server ei salvesta seda ja seda ei saa taastada. BIP39 mehhanismi didaktiline selgitus, miks täpselt kakskümmend neli sõna ja milline tegelik koorem lasub sellel, kes neid valdab.

Et me üksteisest aru saaksime: Kui unustate oma Gmaili parooli, lähtestab Google selle teie jaoks. Kui kaotate 24 sõna, mis moodustavad krüptograafilise identiteedi, pole kelleltki neid küsida. Asi pole selles, et protseduur on range — asi on selles, et teises otsas pole kedagi. See erinevus ongi kogu asja tuum.

Erinevus parooli ja identiteedi vahel

Parool klassikalises internetimudelis ei ole kasutaja identiteet. See on tõend. Kasutajal on identiteet — nimi, e-post, kliendinumber — ja selleks, et tõestada serverile, et ta on see, kes ta väidab end olevat, esitab ta parooli, mida server võrdleb salvestatud jäljendiga. Kui jäljendid ühtivad, lubab server sessiooni. Kui parool kaob, jääb kasutaja samaks kasutajaks; see, mille ta kaotab, on tõend, ja selle taastamiseks on olemas protseduur — e-kiri registreeritud aadressile, turvaküsimus.

Krüptograafiline identiteet toimib teisiti. See ei ole tõend, mida keegi võrdleb salvestatud jäljendiga; see *on* täielik matemaatiline saladus iseeneses. Pole oluline, kus see asub — paberil, seadmes või isegi võõras serveris — identiteet eksisteerib tänu oma matemaatikale, mitte sellele, kes seda valideerib. Siin ilmneb omadus, mis sarnaneb sellele, mida nägime artiklis «Mis SHA-256 tegelikult on»: valdamist ei tõestata saladuse näitamisega, vaid selle kasutamisega allkirjastamiseks. Sel viisil loodud allkirja saab igaüks kontrollida avaliku väärtuse abil, mis on matemaatiliselt tuletatud saladusest endast, ilma et oleks vaja teada saladust ja ilma kolmanda osapoole vahenduseta. Kellel on saladus, see on identiteet; kes selle kaotab, lakkab olemast. Otsus on kategooriline: **pole kedagi, kellelt paluda identiteedi tagastamist. Sellist isikut pole olemas, sest tal polnud seda esimesest hetkest peale.**

Mida kujutavad endast kakskümmend neli sõna

Krüptograafilist identiteeti esindab tavaliselt kolmekümne kahe baidine matemaatiline saladus — kakssada viiskümmend kuus bitti. Arv, mida on raske meeles pidada ja veelgi raskem veatult üles kirjutada. Krüptotööstus lahendas selle probleemi 2013. aastal väikese ja elegantse standardiga nimega BIP39: viis esindada neid kahtksada viitkümmend kuut bitti kahekümne nelja sõna jadana, mis on võetud ametlikust kahe tuhande neljakümne kaheksast sõnast koosnevast loendist. Selle taga olev aritmeetika sobitub elegantselt; kes soovib seda üksikasjalikult näha, leiab selle veerust.

Arvestus algab lõpust. Soovime esindada saladuse kahtksada viitkümmend kuut bitti, lisades kaheksa bitti kontrollsummat: kokku kakssada kuuskümmend neli bitti. Kui jagame need kahekümne nelja sõna peale — mis on hallatav arv ülesmärkimiseks ja etteütlemiseks ilma kadudeta —, peab iga sõna andma täpselt üksteist bitti teavet. Ja üksteist bitti on kaks astmel üksteist võimalust ehk kaks tuhat nelikümmend kaheksa. Seetõttu on ametlik BIP39 sõnavara just selle suurusega: loend on loodud probleemi mõõtude järgi, mitte vastupidi.

Arvestus ei ole dekoratiivne. Kui keegi kirjutab kakskümmend kolm sõna õigesti ja eksib kahekümne neljandaga, tuvastab kontrollsumma selle: tarkvara ütleb talle „see jada ei ole kehtiv“. Kui keegi kirjutab kõik

kakskümmend neli õigesti, tuletab tarkvara üheselt mõistetavalt sama identiteedi. Sõnaloendi valik on samuti kaalutletud: BIP39 sõnavara sõnad on lühikesed, üksteisest erinevad, ilma diakriitiliste märkideta, valitud foneetiliste ja õigekirja segaduste minimeerimiseks. See on sõnavara, mis on loodud selleks, et inimesed saaksid seda kadudeta meeles pidada, kirjutada ja ette öelda.

Fraasist võtmeni

Need kakskümmend neli sõna ei ole kryptograafiline võti, mis sõnumeid allkirjastab. Need on taastatav esitus algsest entroopiast, mis deterministliku protsessi PBKDF2 kaudu muudetakse kuuekümne nelja baidiseks seemneks (seed). Sellest seemnest tuletatakse, samuti deterministlikult, konkreetsed kryptograafilised võtmed, mida kasutaja kasutab: privaatvõti allkirjastamiseks ja vastav avalik võti, mis avaldatakse allkirjade kontrollimiseks. Sama mehhanism erinevates süsteemides: krüptorahad kasutavad secp256k1 kõverat; Signal-protokoll ja paljud kaasaegsed süsteemid kasutavad Ed25519 Curve25519 kõveral. Konkreetse kõvera nagu Ed25519 puhul võtavad BIP32 ja SLIP-0010 standardid selle kuuekümne nelja baidise seemne ja tuletavad deterministlikult kolmkümmend kaks baiti, mis moodustavad tegeliku allkirjastamisvõtme — samad kolmkümmend kaks baiti, millega algab järgmise jaotise koodinäide.

See on standardne viis, kuidas kogu tööstus mehhanismi kasutajale esitleb —krüptorahakotid, detsentraliseeritud identiteedihaldurid, Signal oma püsiva identiteedi osas, Solo2 nende hulgas—: kasutaja ei näe praktikas kunagi seemet ega tuletatud võtmeid. Ta näeb identiteedi loomisel neid kahtkümmet nelja sõna ja kirjutab need soovi korral paberile. Sõnad liiguvad seejärel tema seadmete vahel, kui ta soovib identiteeti migreerida: ta sisestab need uude rakendusse, rakendus tuletab sama seemne, samad võtmed, sama identiteedi. See on kaasaskantav, krüptograafiliselt kindel ja mõistlikkuse piires meeldejäetav mehhanism.

Kuidas võtmega allkirjastada (pintslitõmme Zig-is)

Zig-is mahub sõnumi allkirjastamine Ed25519-ga vaid mõnele reale, kui kakskümmend neljast sõnast tuletatud kolmekümne kahe baidine seeme on olemas:

```
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);
```

Allkirjastamistehing tekitab kuuskümmend neli baiti —mida nimetatakse allkirjaks—, mida sai genereerida ainult vastavast privaatvõtmest. Kontrollimine on avalik: igaüks, kellel on avalik võti, saab kontrollida, kas allkiri vastab sõnumile. Ilma privaatvõtmeta ei saa keegi sellele sõnumile kehtivat allkirja luua; avaliku võtme abil saavad kõik tuvastada, kas allkiri on kehtiv. See asümmeetria võimaldab allkirjastajal tõendada autorlust saladust jagamata.

Eelmine näide on manuaali minimaalne versioon. Solo2 tegelikus koodis läbib ahel kahte faili: üks JavaScriptis, mis asub kasutaja brauseris ja rekonstrueerib entroopia kahekümne neljast sõnast, ja teine Zig-is *zcatcrypto* teegis, mis võtab selle entroopia ja tuletab konkreetsed krüptograafilised võtmed. Alustades brauseri poolelt:

```
// 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 };
}
```

Need kolmkümmend kaks baiti entroopiat koos teise kolmekümne kahega, mis tuletati samas etapis, rändavad Zigi WebAssembly moodulisse, mis genereerib tegelikud Ed25519 võtmed. Täielik funktsioon koos selle lõpliku mälupuhastusega mahub ühele ekraanile:

```
// 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;
}
```

Tasub märkida kahte detaili. Esimene: sama seeme (seed) toodab alati sama võtmepaari — just see võimaldab identiteedi taastamist, sisestades kakskümmend neli sõna uude seadmesse. Teine: seeme kustutatakse viimasel real mälust selgesõnaliselt. Pärast seda punkti ei saaks isegi funktsioon ise võtmeid rekonstrueerida; kasutaja sõnad oleksid ainus allikas.

Neile, kes soovivad seda väikeste arvudega kontrollida. Allkirjastamisskeemi saab läbida tervikuna arvudega, mis on piisavalt väikesed, et arvutusi käsitsi teha. Need, kes eelistavad aritmeetikasse mitte süveneda, võivad selle ploki vahele jätta, ilma et nad kaotaksid artikli lõime; need, kes soovivad näha mehhanismi samm-sammult töötamas, leiavad selle siit. **Avalikud reeglid**, mida igaüks saab lugeda: algarv $p = 23$ (päris Ed25519-s on see umbes seitsmekümne seitsme kohaline; me kasutame kahekümne kolme, et arvutused mahuksid ühele lehele), baas $g = 2$, mille järjekord selles rühmas on $q = 11$, ja konventsioon, et kogu aritmeetika g-ga tehakse *módulo p* ja kõik eksponendid taandatakse *módulo q*. **Privaatne valik**, ainus ja seda ei jagata kunagi: saladus $x = 6$. See ongi identiteet.

Samm 1 — Identiteedi avalik osa. See arvutatakse üks kord ja avaldatakse avalikult.

$y = g^x \bmod p$

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

Identiteedi avalik osa on **18**. Igaüks võib selle võtta ja kasutada seda selle identiteediga tehtud allkirjade kontrollimiseks. Keegi ei saa ainult numbrit 18 jälgides taastada saladust 6: see on diskreetse logaritmi probleem, mille juurde pöördume lõpus.

Samm 2 — Sõnumi allkirjastamine. Identiteedi omanik soovib allkirjastada sõnumit $m = 7$. Ta alustab uue juhusliku väärtuse $k = 4$ valimisega, mida kasutatakse ainult üks kord ja mida ei jagata kunagi (päris Ed25519-s tuletatakse k deterministlikult sõnumist ja saladusest, et vältida korduskasutamise ohtu, kuid selle roll on täpselt selline). Seejärel arvutab ta kolm arvu:

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

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

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

Allkiri on paar **(r, s) = (16, 10)**. See rändab avalikult koos sõnumiga. Igaüks saab seda lugeda. Didaktiline märkus: päris Ed25519-s on funktsioon H SHA-512, mis on krüptograafiliselt robustne; siin kasutame lihtsustust $e = (r + m) \bmod q$, et lugeja saaks sammud läbida ilma hashi arvutamata. Algoritmi struktuur on sama.

Samm 3 — Allkirja kontrollimine. Kontrollijal on avalik osa $y = 18$, sõnum $m = 7$ ja allkiri $(r, s) = (16, 10)$. Ta rekonstrueerib e samal viisil — $e = (16 + 7) \bmod 11 = 1$ — ja kontrollib, kas see võrdsus kehtib:

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

Arvutab mõlemad pooled eraldi:

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

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

Mõlemad pooled annavad tulemuseks **12**. Allkiri on kehtiv. Igaüks, kellel on avalik osa 18, võib jõuda sellele järeldusele, teadmata kunagi, et saladus oli 6.

Aga kolmas osapool, kes püüab võltsida? Eva on näinud kõike avalikku kanalit pidi liikumas: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Selle identiteedi nimel *teistsuguse* sõnumi allkirjastamiseks peaks ta teadma x-i. Tema ainus viis on küsida endalt: "millise eksponendi x puhul kehtib $2^x \bmod 23 = 18$?". Kui $p = 23$, saab ta proovida 0, 1, 2, 3, … ja leida selle sekunditega. Kuid asendades 23 algarvuga, mis vastab Ed25519 tegelikele mõõtmetele, ületab võimalike eksponentide arv nähtavas universumis olevate aatomite arvu. **Tänapäeval ei ole inimkonnale teada ühtegi algoritmi, mis suudaks selle ruumi läbida vähem kui miljardite aastatega.** See on sama diskreetse logaritmi probleem, mis on eelmise artikli Diffie-Hellmani aluseks, rakendatuna siin allkirjastamisskeemile.

See, mille me just läbisime, on *täpselt* Schnorr, allkirjastamisskeem, mille variant on elliptilisele kõverale kohandatud Ed25519. Päris Ed25519-s tehakse kõik toimingud konkreetse kõvera (Curve25519) punktidega, mitte täisarvudega modulo algarv, ja funktsioon H on SHA-512 ülaltoodud mängusumma asemel. Need kaks asendust on rakenduslikud kohandused — krüptograafilise vastupidavuse saavutamine jõumeetodil rünnakutele ja täiendavate turvaomaduste saavutamine k jaoks. Algoritmiline struktuur, kolm toimingut ja asümmeetria põhjus on samad.

Siinkohal on sobiv teha lühike peatus, sest kogu ahelat võib kergel pilgul segi ajada kolmiku teise primitiiviga: hashiga. See ei ole nii. Hash on unikaalne funktsioon, mis surub kokku — siseneb palju baite, väljub lühike jäljend, sellega tee lõpeb. Krüptograafiline identiteet on matemaatiliselt täiendav paar: saladus jääb ja allkirjastab; selle avalik vastaspool avaldatakse ja kontrollitakse. Kus hash kollabeerib teabe ühes suunas, seal seab identiteet kahe poole vahel asümmeetria. Hash tõendab, mida öeldi; identiteet tõendab, kes seda ütles.

Mida fraas ei ole

Tuleb selgitada kolme sagedast eksiarvamust. Fraas ei ole parool selle õiges tähenduses: seda ei võrrelda serverisse salvestatud sõrmejäljega; see sisestatakse kasutaja seadmesse identiteedi matemaatiliseks taastamiseks. Fraasi ei saa taastada: kui see kaob, pole kelleltki seda küsida; kui seda dubleeritakse, dubleeritakse ka identiteet. Fraas ei ole identiteedist eraldatav volikiri: fraas *on* identiteet. Kes seda valdab, saab selle nimel tegutseda ilma täiendava loata, ilma autoriseerimisprotsessita, ilma taastamisvõimaluseta.

Just see kolmas omadus muudab asja kaalu. Kaotatud parool on administratiivne ebamugavus. Kaotatud kryptograafiline identiteet ongi identiteet. Kolmandate isikute poolt leitud paber fraasiga ei ole konto varguse oht: see on kogu identiteedi üleandmine. Süsteemi lubadusega —et keegi ei saaks teie identiteeti tühistada ega teid suvaliselt blokeerida— kaasneb lahutamatult vastutus —et te olete ainsaks hoidjaks millegi eest, mida keegi teine ei saa teie eest taastada.

Lubadus ja kaal

Krüptograafilist identiteedimudelit nimetatakse sageli *suveräänidentiteediks* —self-sovereign ingliskeelses kirjanduses—. Sõnavalik on teadlik ja kirjeldab olukorda üsna täpselt. Kasutaja on oma identiteedi üle suverään peaaegu keskaegses tähenduses: seda ei anna ükski kuningas, ükski väljastaja ega ükski keskasutus; samuti ei saa keegi neist seda tühistada. Kuid samamoodi nagu keskaegne monarh, kannab kasutaja oma vigade eest täielikku vastutust: pole regenti, kes teeks tema eest otsuseid, kui ta pitsati kaotab.

Valikul kolmanda osapoole hallatava identiteedi ja suveräänidentiteedi vahel ei ole ühtset universaalset õiget vastust. Ebaolulise foorumikonto puhul on hallatav identiteet tõenäoliselt riskiga proportsionaalne. Juriidiliselt siduvaid dokumente allkirjastava professionaalse identiteedi, oma sääste valvava majandusliku identiteedi või tundlikku teavet usaldanud klientidega toimuva professionaalse suhtluse identiteedi puhul on lugu teine. Seal lakkab küsimus olemast „kas see on mugav?“ ja muutub küsimuseks „kellel peale minu on võim tegutseda minu nimel ja millistel asjaoludel?“.

Kus see mehhanism reaalsetes süsteemides ilmneb

BIP39 sündis Bitcoin maailmas 2013. aastal ja levis kiiresti kogu krüptovaluutade ökosüsteemi: iga tõsine rahakott aktsepteerib täna kaheteistkümne- või kahekümne nelja sõnalist BIP39 fraasi oma valdaja majandusliku identiteedi varukoopiana. Väljaspool krüptovaluutasid ilmub sama aluskontseptsioon — krüptograafiline paar, mis tõendab autorlust ilma vahendajata — teistes süsteemides erineva süntaksiga. SSH-võtmed, mida süsteemiadministraator kasutab oma serveritele juurdepääsuks, on klassikaline juhtum: privaatvõti, mida administraator hoiab oma masinas, ja avalik võti, mis kopeeritakse igasse serverisse; ükski tsentraliseeritud teenusega võrreldav üksus ei sekku. Signali protokoll kasutab seadmes püsivat Ed25519 võtmematerjali; Euroopa eIDAS toetub oma kvalifitseeritud elektroonilise allkirja osas samale krüptograafilisele põhimõttele, selle vahega, kad võtit hoiab kasutaja asemel kvalifitseeritud usaldusteenuse pakkuja.

Solo2, selle väljaande kirjastusplatvorm, kasutab iga kasutaja identiteedina kahekümne nelja sõnalist BIP39 fraasi. Kasutaja näeb sõnu üks kord konto loomisel. Neid ei salvestata Solo2 ega kellegi teise serverisse: kui kasutaja need üles märgib ja neid hoiab, säilitab ta oma identiteedi igavesti. Kui ta need kaotab, siis ta kaotab need. See on loogiline tagajärg arhitektuurile, kus puudub vahendav operaator: kui Solo2 saaks identiteedi tagastada selle kaotanud kasutajale, saaks ta selle anda ka kelle iganes, kes avaldab Solo2-le survet selle saamiseks.

Professionaalsele lugejale

Neli kaalutlust neile, kes kaaluvad krüptograafilise isesuveräänse (autosoberana) identiteedi kasutuselevõttu professionaalses kontekstis:

1. Fraas on identiteet. Füüsiline hoiustamine — paber, mitu koopiat eri kohtades, lõpuks pikaajaliseks kasutamiseks graveeritud metall — pakub rohkem garantiisid kui digitaalne hoiustamine, mis suurendab ründepinda ilma kaotusriski vähendamata.
2. Taastamist ei ole. Protsessi kavandamine eeldusega, ka ühel päeval esmane koopia kaob, on palju mõistlikum kui selle avastamine kaotamise päeval. Teine geograafiliselt eraldatud koopia lahendab peaaegu kõik stsenaariumid.
3. See ei ole sama mis eIDAS-e kvalifitseeritud sertifikaat. Liidu kvalifitseeritud allkirja puhul — notariaalaktid, teatud toimingud ametiasutustega — nõuavad õigusaktid kvalifitseeritud pakkujat, kes hoiab võtit. Krüptograafiline isesuveräänne identiteet teenib professionaalset suhtlust ja tõendusväärtusega dokumentide allkirjastamist, kuid ei asenda automaatselt kvalifitseeritud sertifikaati juhtudel, kui norm seda nõuab.
4. Kui identiteet kuulub üleandmisele — pärimine, professionaalne järjepidevus, tegevuse lõpetamine —, on soovitatav protseduur ette valmistada varem, mitte hiljem. Formaalsed protseduurid pitsatvahaga (lacre) suletud ümbrikega, juhised testamenditäitjale, hoiustamine notaribüroos on klassikalised kokkulepped, mis sobivad ideaalselt vara krüptograafilise olemusega.

See artikkel lõpetab kontseptuaalse trio, mis tsükli avas — hash, krüpteerimine, identiteet —. Need kolm ideed toetuvad üksteisele: hash annab muutumatu jälje, krüpteerimine annab konfidentsiaalsuse ilma usaldusväärse kolmanda osapooleta, identiteet annab autorluse ilma luba andva kolmanda osapooleta. Kõigil kolmel on omadus, mis ei ole samuti ideoloogiline: nad annavad teenuse haldajalt kasutajale üle tehnilised võimalused, mis traditsiooniliselt kuulusid operaatorile. Koos nendega antakse üle ka vastutus. Aus rääkimine neist kolmest nõuab rääkimist ka ülejäänud kahest.

Allikad ja täiendav lugemine

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, 2013. aasta Bitcoin täiendusettepanek. Krüptotööstuse taastefraaside de facto standard.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), sh Ed25519. IETF, jaanuar 2017. Suures osas kaasaegses tööstuses kasutatava allkirjastamisskeemi normatiivne spetsifikatsioon.

- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, versioon 2.0. IETF, september 2000. Määratleb BIP39 fraasist seemneks (seed) tuletamisel kasutatava PBKDF2 algoritmi.
- Määrus (EL) 910/2014 (eIDAS) ja selle areng määrusega (EL) 2024/1183 (eIDAS 2) — Euroopa e-identimise ja kvalifitseeritud allkirjade raamistik. Teistsugune režiim kui isesuveräänne, kuid kontseptuaalselt toetub samadele krüptograafilistele primitiividele.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Kanooniline tekst isesuveräänse mudeli põhimõtetest ja kohustustest, varasem, kuid asjakohane tänapäevaste lahenduste pere mõistmiseks.

← EelmineÄrimudel kui usalduse signaalJärgmine →Self-hosting kui professionaalne praktika

Viimased lugemised

- Mõtisklus · 29. juuni 2026 Sa ei ole anonüümne
- Refleksioon · 27. mai 2026 Mida allkiri ei saa parandada
- Analüüs · 26. mai 2026 Tõeline vs näiline privaatsus: küsimused, mida tasub endalt küsida

Võtke see artikkel endaga kaasa kuhu iganes vaja.

↓ Markdown ↓ Lihttekst ↓ PDF

Fail laaditakse alla teie seadmesse. Sealt saate selle salvestada, importida Solo2-sse või jagada kus iganes soovite. Cuadernos ei otsusta sihtkohta teie eest.

Lakipitsat · SHA-256 5e02ac5ea18e234fe539a0b2b39f54ce373126893cf82f987be6684899cca3f9

Cuadernos Lacre · Ettevõtte Menzuri Gestión S.L. väljaanne ·
kirjutanud R.Eugenio · toimetanud Solo2 meeskond.

See veebileht ei kasuta küpsiseid. Kõik, mida sinu brauser laadib, on meie kirjutatud või meie järelevalve all ja paikneb meie Euroopa serverites: anonüümne külastuste loendur (Umami, ise majutatud) ja minimaalne vajalik JavaScript keelevaliku ning hele või tume teema eelistuse jaoks, mis salvestatakse sinu enda seadmesse. Ei mingeid väliste ettevõtete ressursse, ei mingeid jälitajaid, ei mingit profiilianalüüsi, ei mingit andmete jagamist. Kui soovid meid jälgida: RSS.