De 24 woorden: wat een cryptografische identiteit is

Een cryptografische identiteit is geen wachtwoord: geen enkele server bewaart het en het kan niet worden hersteld. Een didactische uitleg van het BIP39-mechanisme, waarom precies vierentwintig woorden en welk reëel gewicht er rust op degene die ze bezit.

Om elkaar goed te begrijpen: als je je Gmail-wachtwoord vergeet, stelt Google het voor je opnieuw in. Als je de 24 woorden verliest die een cryptografische identiteit vormen, is er niemand aan wie je ze kunt vragen. Het is niet zo dat de procedure streng is — het punt is dat er niemand aan de andere kant is. Dat verschil is het hele eieren eten.

Het verschil tussen een wachtwoord och een identiteit

Een wachtwoord in het klassieke internetmodel is niet de identiteit van de gebruiker. Het is een bewijsstuk. De gebruiker heeft een identiteit — een naam, een e-mailadres, een klantnummer — en om aan een server te bewijzen dat hij is wie hij zegt te zijn, presenteert hij een wachtwoord dat de server vergelijkt met een opgeslagen vingerafdruk. Als de vingerafdrukken overeenkomen, verleent de server de sessie. Als het wachtwoord verloren gaat, blijft de gebruiker dezelfde gebruiker; wat hij verliest is het bewijsstuk, en er bestaat een herstelprocedure — een e-mail naar het geregistreerde adres, een veiligheidsvraag — om het te herstellen.

Een cryptografische identiteit werkt anders. Het is geen inloggegeven dat iemand vergelijkt met een opgeslagen vingerafdruk; het *is* een volledig wiskundig geheim op zich. Het maakt niet uit waar het zich bevindt — op een papiertje, in een apparaat, zelfs op een server van derden — de identiteit bestaat door zijn wiskunde, niet door wie het valideert. Hier verschijnt een eigenschap die lijkt op wat we zagen in «Wat SHA-256 echt is»: bezit wordt niet bewezen door het geheim te tonen, maar door het te gebruiken om te ondertekenen. De op deze manier geproduceerde handtekening kan door iedereen worden gecontroleerd met een publieke waarde die wiskundig is afgeleid van het geheim zelf, zonder dat men het geheim hoeft te kennen en zonder dat een derde partij bemiddelt bij de controle. Wie het geheim heeft, is de identiteit; wie het verliest, houdt op dat te zijn. Het vonnis is onverbiddelijk: **er is niemand aan wie je kunt vragen om je de identiteit terug te geven. Die persoon bestaat niet, omdat hij die in de eerste plaats niet had.**

Wat vierentwintig woorden vertegenwoordigen

De cryptografische identiteit wordt gewoonlijk gerepresenteerd door een wiskundig geheim van tweeëndertig bytes — tweehonderdzesenvijftig bits. Een getal dat moeilijk te onthouden is en nog moeilijker foutloos over te schrijven. De cryptosector loste dit probleem in 2013 op met een kleine en elegante standaard genaamd BIP39: een manier om die tweehonderdzesenvijftig bits weer te geven als een reeks van vierentwintig woorden uit een officiële lijst van tweeduizend achtenveertig. De achterliggende rekenkunde sluit elegant aan; wie het in detail wil zien, vindt het in de kantlijn.

De berekening begint bij het einde. We willen de tweehonderdzesenvijftig bits van het geheim weergeven door acht bits checksum toe te voegen: tweehonderdvierenzeventig bits in totaal. Als we deze verdelen over vierentwintig woorden — een hanteerbaar aantal om zonder verlies te noteren en te dicteren — moet elk woord precies elf bits informatie leveren. En elf bits zijn twee tot de elfde macht mogelijkheden, oftewel tweeduizend

achtenveertig. Vandaar dat de officiële BIP39-woordenschat precies die omvang heeft: de lijst is gemaakt op maat van het probleem, niet andersom.

De berekening is niet decoratief. Als iemand drieëntwintig woorden correct overschrijft en een fout maakt bij het vierentwintigste, zal de checksum dit detecteren: de software zal hem vertellen "deze reeks is niet geldig". Als iemand alle vierentwintig woorden correct overschrijft, zal de software ondubbelzinnig dezelfde identiteit afleiden. De keuze van de woordenlijst is ook weloverwogen: de woorden in het BIP39-vocabulaire zijn kort, onderling verschillend, zonder diakritische tekens, gekozen om fonetische en spellingverwarring te minimaliseren. Het is een vocabulaire dat is ontworpen om door mensen zonder verlies te worden onthouden, geschreven en gedicteerd.

Van de frase naar de sleutel

De vierentwintig woorden zijn niet de cryptografische sleutel die berichten ondertekent. Ze zijn een herstelbare weergave van de oorspronkelijke entropie die, via een deterministisch proces genaamd PBKDF2, wordt getransformeerd in een seed van vierenzestig bytes. Van die seed worden, eveneens deterministisch, de specifieke cryptografische sleutels afgeleid die de gebruiker gebruikt: een privésleutel om te ondertekenen und een bijbehorende publieke sleutel die wordt gepubliceerd om de handtekeningen te verifiëren. Hetzelfde mechanisme in verschillende systemen: cryptocurrencies gebruiken de secp256k1-curve; het Signal-protocol en veel moderne systemen gebruiken Ed25519 op de Curve25519-curve. Voor een specifieke curve zoals Ed25519 nemen de BIP32- en SLIP-0010-standaarden die seed van vierenzestig bytes en leiden ze op deterministische wijze de tweeëndertig bytes af die de effectieve ondertekeningssleutel vormen — dezelfde tweeëndertig bytes waarmee het codevoorbeeld in de volgende sectie begint.

Dit is de standaardmanier waarop de hele industrie het mechanisme aan de gebruiker presenteert — cryptocurrency-wallets, beheerders van gedecentraliseerde identiteit, Signal in zijn persistente identiteitsgedeelte, Solo2 onder hen—: de gebruiker ziet in de praktijk nooit de seed of de afgeleide sleutels. Hij ziet de vierentwintig woorden bij het aanmaken van zijn identiteit en noteert ze optioneel op een stuk papier. De woorden reizen vervolgens tussen zijn apparaten wanneer hij de identiteit wil migreren: hij voert ze in de nieuwe applicatie in, de applicatie leidt dezelfde seed, dezelfde sleutels en dezelfde identiteit af. Het is een draagbaar, cryptografisch solide en, binnen redelijke grenzen, onthoudbaar mechanisme.

Hoe te ondertekenen met de sleutel (een Zig-penseelstreek)

In Zig past het ondertekenen van een bericht met Ed25519 in een paar regels, zodra je de van de vierentwintig woorden afgeleide tweeëndertig bytes seed hebt:

```
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);
```

De ondertekeningsoperatie produceert vierenzestig bytes —handtekening genoemd— die alleen gegenereerd konden worden op basis van de bijbehorende privésleutel. De verificatie is openbaar: iedereen met de publieke sleutel kan controleren of de handtekening overeenkomt met het bericht. Zonder de privésleutel kan niemand een

geldige handtekening voor dat bericht produceren; met de publieke sleutel kan iedereen detecteren of een handtekening geldig is. Deze asymmetrie is wat de ondertekenaar in staat stelt het auteurschap aan te tonen zonder het geheim te delen.

Het vorige voorbeeld is de minimale handleidingsversie. In de echte Solo2-code doorloopt de keten twee bestanden, één in JavaScript dat in de browser van de gebruiker leeft en de entropie reconstrueert uit de vierentwintig woorden, een ander in Zig binnen de *zcatcrypto*-bibliotheek die die entropie neemt en de specifieke cryptografische sleutels afleidt. Beginnend aan de kant van de browser:

```
// 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 };
}
```

Die tweeëndertig bytes entropie reizen, samen met nog eens tweeëndertig afgeleid in dezelfde stap, naar de WebAssembly-module van Zig die de eigenlijke Ed25519-sleutels genereert. De volledige functie, met zijn uiteindelijke geheugenopschoning, past op één scherm:

```
// 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;
}
```

Twee details zijn vermeldenswaard. Ten eerste: eenzelfde seed produceert altijd hetzelfde sleutelpaar — het is precies dat wat identiteitsherstel mogelijk maakt door de vierentwintig woorden op een nieuw apparaat in te voeren. Ten tweede: de seed wordt in de laatste regel expliciet uit het geheugen gewist. Na dat punt zou zelfs de functie zelf de sleutels niet meer kunnen reconstrueren; de woorden van de gebruiker zouden de enige bron zijn.

Voor wie het met kleine getallen wil controleren. Het ondertekeningsschema kan volledig worden doorlopen met cijfers die klein genoeg zijn om de berekeningen met de hand te doen. Wie liever niet in de rekenkunde duikt, kan dit blok overslaan zonder de draad van het artikel te verliezen; wie het mechanisme stap voor stap aan het werk wil zien, vindt het hier. **De openbare regels**, die iedereen kan lezen: een priemgetal $p = 23$ (in echte Ed25519 is het ongeveer zevenenzeventig cijfers lang; we gebruiken drieëntwintig zodat de berekeningen op één pagina passen), een basis $g = 2$ waarvan de orde in deze groep $q = 11$ is, en de conventie dat alle rekenkunde met *g módulo p* wordt gedaan en alle exponenten *módulo q* worden gereduceerd. **De privékeuze**, slechts één en nooit gedeeld: het geheim $x = 6$. Dat is de identiteit.

Stap 1 — Het openbare deel van de identiteit. Het wordt eenmalig berekend en openlijk gepubliceerd.

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

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

Het openbare deel van de identiteit is **18**. Iedereen kan het nemen en gebruiken om handtekeningen te verifiëren die met deze identiteit zijn gezet. Niemand kan, door alleen de 18 te observeren, het geheim 6 herstellen: dat is het probleem van de discrete logaritme waar we aan het eind op terugkomen.

Stap 2 — Een bericht ondertekenen. De houder van de identiteit wil bericht $m = 7$ ondertekenen. Hij begint met het kiezen van een nieuwe willekeurige waarde $k = 4$, die slechts één keer wordt gebruikt en nooit wordt gedeeld (in echte Ed25519 wordt *k* deterministisch afgeleid van het bericht en het geheim om het gevaar van hergebruik te vermijden, maar de rol die het speelt is precies dit). Daarna berekent hij drie getallen:

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

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

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

De handtekening is het paar **(r, s) = (16, 10)**. Het reist openlijk mee met het bericht. Iedereen kan het lezen. Didactische opmerking: in echte Ed25519 is de functie *H* SHA-512, cryptografisch robuust; hier gebruiken we de vereenvoudiging $e = (r + m) \bmod q$ zodat de lezer de stappen kan doorlopen zonder een hash te hoeven berekenen. De structuur van het algoritme is hetzelfde.

Stap 3 — De handtekening verifiëren. De verifieerder heeft het openbare deel $y = 18$, het bericht $m = 7$, en de handtekening $(r, s) = (16, 10)$. Hij reconstrueert e op dezelfde manier — $e = (16 + 7) \bmod 11 = 1$ — en controleert of deze gelijkheid opgaat:

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

Berekent de twee kanten afzonderlijk:

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

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

Beide kanten geven **12**. De handtekening is geldig. Iedereen met het openbare deel 18 kan tot deze conclusie komen zonder ooit te hebben geweten dat het geheim 6 was.

En een derde partij die probeert te vervalsen? Eva heeft alles wat openbaar is door het kanaal zien gaan: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Om een *ander* bericht te ondertekenen in naam van deze identiteit, zou ze x moeten kennen. Haar enige weg is zich afvragen: «voor welke exponent x geldt $2^x \bmod 23 = 18$?». Met $p = 23$ kan ze 0, 1, 2, 3, … proberen en het binnen enkele seconden vinden. Maar door 23 te vervangen door een priemgetal van de reële afmetingen van Ed25519, overstijgt de ruimte van mogelijke exponenten het aantal atomen in het waarneembare universum. **Er bestaat vandaag de dag geen algoritme dat bekend is bij de mensheid dat die ruimte in minder dan miljarden jaren kan doorlopen.** Het is hetzelfde probleem van de discrete logaritme dat de basis vormt voor de Diffie-Hellman uit het vorige artikel, hier toegepast op het ondertekeningsschema.

Wat we zojuist hebben doorlopen is *precies* Schnorr, het ondertekeningsschema waarvan Ed25519 een variant is die is aangepast aan een elliptische curve. In echte Ed25519 worden alle bewerkingen uitgevoerd op de punten van een specifieke curve (Curve25519) in plaats van op gehele getallen modulo een priemgetal, and de functie H is SHA-512 in plaats van de speelgoedsom die we hierboven gebruikten. De twee vervangingen zijn implementatie-aanpassingen — het verkrijgen van cryptografische weerstand tegen brute force, het verkrijgen van aanvullende beveiligingseigenschappen voor k —. De algoritmische structuur, de drie bewerkingen, de reden voor de asymmetrie, zijn hetzelfde.

Een kort verblijf is hier op zijn plaats, omdat de hele keten bij een snelle blik verward kan worden met een andere primitieve van het trio: de hash. Dat is het niet. Een hash is een unieke functie die comprimeert — er gaan veel bytes in, er komt een korte vingerafdruk uit, daar eindigt de weg. Een cryptografische identiteit is een complementair wiskundig paar: het geheim blijft en ondertekent; zijn openbare tegenhanger wordt gepubliceerd en verifieert. Waar de hash informatie in één richting doet instorten, vestigt de identiteit een asymmetrie tussen twee helften. De hash getuigt van wat er gezegd is; de identiteit getuigt van wie het zei.

Wat de frase niet is

Drie veelvoorkomende misverstanden moeten worden opgehelderd. De frase is geen wachtwoord in de eigenlijke zin: het wordt niet vergeleken met een op een server opgeslagen vingerafdruk; het wordt in het apparaat van de gebruiker ingevoerd om de identiteit wiskundig te reconstrueren. De frase kan niet worden hersteld: als het verloren gaat, is er niemand aan wie je het kunt vragen; als het wordt gedupliceerd, wordt de identiteit ook gedupliceerd. De frase is geen van de identiteit scheidbaar bewijsstuk: de frase *is* de identiteit. Wie het heeft, kan als die identiteit optreden, zonder aanvullende toestemming, zonder autorisatieproces, zonder mogelijk herstel.

Juist deze derde eigenschap is wat het gewicht van de zaak verandert. Een verloren wachtwoord is een administratief ongemak. Een verloren cryptografische identiteit is de identiteit zelf. Een door derden gevonden papier met de frase is geen risico op accountdiefstal: het is de overdracht van de gehele identiteit. De belofte van het systeem — dat niemand je identiteit kan herroepen of je willekeurig kan blokkeren — gaat onafscheidelijk gepaard met de verantwoordelijkheid — dat jij de enige bewaarder bent van iets dat niemand voor jou kan herstellen.

De belofte und het gewicht

Het model van cryptografische identiteit krijgt vaak het predicaat *zelf-soeverein* —self-sovereign in de Engelstalige literatuur—. De woordkeuze is bewust en beschrijft de toestand vrij nauwkeurig. De gebruiker is soeverein over zijn identiteit in een bijna middeleeuwse zin: het wordt door geen enkele koning, geen enkele uitgever, geen enkele centrale autoriteit verleend; noch kan het door een van de voorgaanden worden ingetrokken. Maar ook draagt de gebruiker, net als de middeleeuwse monarch, de volledige consequentie van zijn fouten: er is geen regent die in zijn plaats beslissingen neemt als hij het zegel verliest.

De keuze tussen een door een derde beheerde identiteit en een zelf-soevereine identiteit heeft geen universeel juist antwoord. Voor een irrelevant forumaccount is de beheerde identiteit waarschijnlijk proportioneel aan het risico. Voor een professionele identiteit die juridisch bindende documenten ondertekent, voor een economische identiteit die eigen spaargeld bewaakt, voor een professionele communicatie-identiteit met klanten die gevoelige informatie hebben toevertrouwd, verandert de zaak. Daar stopt de vraag «is het handig?» te zijn en wordt het «wie, behalve ikzelf, heeft de macht om als mij op te treden, en onder welke omstandigheden?».

Waar dit mechanisme in echte systemen verschijnt

BIP39 werd in 2013 geboren in de wereld van Bitcoin en verspreidde zich snel naar het hele ecosysteem van cryptovaluta: elke serieuze wallet accepteert tegenwoordig een BIP39-zin van twaalf of vierentwintig woorden als back-up van de economische identiteit van de houder. Buiten cryptovaluta verschijnt hetzelfde onderliggende concept — een cryptografisch paar dat auteurschap bewijst zonder tussenpersoon — in andere systemen met een andere syntaxis. De SSH-sleutels die een systeembeheerder gebruikt om toegang te krijgen tot zijn servers zijn een klassiek geval: een privésleutel die de beheerder op zijn machine bewaart en een openbare die naar elke server wordt gekopieerd; er komt geen entiteit tussenbeide die vergelijkbaar is met een gecentraliseerde dienst. Het Signal-protocol gebruikt Ed25519 met persistent sleutelmateriaal op het apparaat; het Europese eIDAS rust, in zijn deel over de gekwalificeerde handtekening, op hetzelfde cryptografische principe, met het verschil dat de sleutel wordt bewaard door een gekwalificeerde vertrouwensdienstverlener in plaats van de gebruiker.

Solo2, het uitgeversplatform van deze publicatie, gebruikt een BIP39-zin van vierentwintig woorden als de identiteit van elke gebruiker. De gebruiker ziet de woorden één keer bij het aanmaken van zijn account. Ze worden op geen enkele server van Solo2 of van wie dan ook opgeslagen: als de gebruiker ze noteert en bewaart, behoudt hij zijn identiteit voor altijd. Als hij ze verliest, verliest hij ze. Het is het logische gevolg van een architectuur zonder operator in het midden: als Solo2 de identiteit zou kunnen teruggeven aan de gebruiker die deze heeft verloren, zou het deze ook kunnen geven aan iedereen die Solo2 onder druk zet om het te geven.

Voor de professionele lezer

Vier overwegingen voor degenen die de adoptie van een cryptografische zelf-soevereine (autosoberana) identiteit in een professionele context overwegen:

1. De zin is de identiteit. Fysieke bewaring — papier, meerdere kopieën op verschillende plaatsen, uiteindelijk gegraveerd metaal voor langdurig gebruik — biedt meer garanties dan digitale bewaring, die het aanvalsoppervlak vergroot zonder het risico op verlies te verkleinen.
2. Er is geen herstel mogelijk. Het proces ontwerpen vanuit de veronderstelling dat de primaire kopie op een dag verloren gaat, is veel verstandiger dan het ontdekken op de dag dat deze verloren gaat. Een tweede geografisch gescheiden kopie lost bijna alle scenario’s op.
3. Het is niet hetzelfde als een gekwalificeerd eIDAS-certificaat. Voor gekwalificeerde handtekeningen in de Unie — notariële akten, bepaalde procedures bij de overheid — vereist de wetgeving een gekwalificeerde aanbieder die de sleutel bewaart. Cryptografische zelf-soevereine identiteit dient voor professionele communicatie en documentaire ondertekening met bewijskracht, maar vervangt niet automatisch het gekwalificeerde certificaat in gevallen waarin de norm dit vereist.
4. Als de identiteit moet worden overgedragen — erfenis, professionele opvolging, beëindiging van de activiteit — is het raadzaam om de procedure vooraf voor te bereiden, niet achteraf. Formele procedures met enveloppen die verzegeld zijn met zegellak (lacre), instructies aan een executeur-testamentair, depot

bij een notariskantoor, zijn klassieke regelingen die perfect compatibel zijn met de cryptografische aard van het actief.

Dit artikel sluit het conceptuele trio af dat de cyclus opende — hash, versleuteling, identiteit —. De drie ideeën bouwen op elkaar voort: de hash geeft de onveranderlijke vingerafdruk, de versleuteling geeft de vertrouwelijkheid zonder vertrouwde derde partij, de identiteit geeft het auteurschap zonder verlenende derde partij. De drie delen een eigenschap die ook niet ideologisch is: ze dragen technische mogelijkheden die traditioneel bij de operator lagen, over van degene die een dienst beheert naar degene die deze gebruikt. Ze dragen daarmee ook verantwoordelijkheden over. Eerlijk spreken over elk van de drie vereist ook spreken over de andere twee.

Bronnen und verdere lectuur

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, Bitcoin-verbeteringsvoorstel uit 2013. De facto standaard voor herstelzinnen in de crypto-industrie.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), inclusief Ed25519. IETF, januari 2017. Normatieve specificatie van het handtekeningschema dat in een groot deel van de hedendaagse industrie wordt gebruikt.
- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, versie 2.0. IETF, september 2000. Definieert het PBKDF2-algoritme dat wordt gebruikt in de BIP39-afleiding van zin naar seed.
- Verordening (EU) 910/2014 (eIDAS) und zijn evolutie door Verordening (EU) 2024/1183 (eIDAS 2) — Europees kader voor elektronische identiteit en gekwalificeerde handtekening. Een ander regime dan het zelf-soevereine, maar conceptueel ondersteund door dezelfde cryptografische primitieven.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Canonieke tekst over de principes en toezeggingen van het zelf-soevereine model, ouder maar relevant voor het begrijpen van de familie van hedendaagse oplossingen.

← VorigeHet bedrijfsmodel als signaal van vertrouwenVolgende →Self-hosting als professionele praktijk

Recente artikelen

- Reflectie · 29 juni 2026 Je bent niet anoniem
- Reflectie · 27 mei 2026 Wat een handtekening niet kan oplossen
- Analyse · 26 mei 2026 Echte vs. schijnbare privacy: de vragen die men zich moet stellen

Neem dit artikel mee naar waar u het nodig heeft.

↓ Markdown ↓ Platte tekst ↓ PDF

Het bestand wordt gedownload naar uw apparaat. Van daaruit kunt u het opslaan, importeren in Solo2 of delen waar u maar wilt. Cuadernos beslist niet voor u over de bestemming.

Lakzegel · SHA-256 5c0e69f77e5ca1f18442213ffac714cad384750b6196446733e839bfac64a2b9

Cuadernos Lacre · Een uitgave van Menzuri Gestión S.L. ·
geschreven door R.Eugenio · geredigeerd door het team van Solo2.

wordt opgeslagen. Geen bronnen van derden, geen trackers, geen profilering, geen delen van gegevens. Als u ons wilt volgen: RSS.