24 Kata: Apa itu Identitas Kriptografi

Identitas kriptografi bukanlah kata sandi: tidak ada server yang menyimpannya dan tidak dapat dipulihkan. Penjelasan didaktis tentang mekanisme BIP39, mengapa tepat dua puluh empat kata, dan beban nyata apa yang dipikul oleh mereka yang memilikinya.

Agar kita saling memahami: Jika Anda lupa kata sandi Gmail, Google akan menyetel ulangnya untuk Anda. Jika Anda kehilangan 24 kata yang membentuk identitas kriptografi, tidak ada orang yang bisa dimintai kata-kata tersebut. Bukannya prosedurnya ketat — masalahnya adalah tidak ada orang di ujung sana. Perbedaan itulah yang menjadi segalanya.

Perbedaan Antara Kata Sandi dan Identitas

Kata sandi, dalam model internet klasik, bukanlah identitas pengguna. Itu adalah voucer. Pengguna memiliki identitas —nama, email, nomor pelanggan— dan, untuk membuktikan kepada server bahwa mereka adalah orang yang mereka klaim, mereka memberikan kata sandi yang dibandingkan server dengan jejak yang telah disimpan. Jika jejak tersebut cocok, server mengizinkan sesi tersebut. Jika kata sandi hilang, pengguna tetap menjadi pengguna yang sama; yang hilang adalah voucer, dan ada prosedur pemulihan —email ke alamat terdaftar, pertanyaan keamanan— untuk memulihkannya.

Identitas kriptografi bekerja dengan cara yang berbeda. Ini bukan kredensial yang dibandingkan seseorang dengan jejak yang disimpan; ini *adalah* rahasia matematis yang lengkap dalam dirinya sendiri. Tidak masalah di mana ia berada —di kertas, di perangkat, bahkan di server asing—: identitas itu ada karena matematikanya, bukan karena siapa yang memvalidasinya. Di sini muncul properti yang mirip dengan yang kita lihat di «Apa itu SHA-256 sebenarnya»: kepemilikan tidak dibuktikan dengan menunjukkan rahasia, melainkan dengan menggunakannya untuk menandatangani. Tanda tangan yang dihasilkan dengan cara ini dapat diperiksa oleh siapa pun dengan nilai publik yang diturunkan secara matematis dari rahasia itu sendiri, tanpa perlu mengetahui rahasianya, dan tanpa pihak ketiga yang menengahi pemeriksaan tersebut. Siapa pun yang memiliki rahasia tersebut, dialah identitasnya; siapa pun yang kehilangannya, berhenti menjadi identitas tersebut. Putusannya mutlak: **tidak ada orang yang bisa dimintai untuk mengembalikan identitas tersebut kepada Anda. Orang itu tidak ada, karena mereka tidak memilikinya sejak awal.**

Apa yang Direpresentasikan oleh Dua Puluh Empat Kata

Identitas kriptografi biasanya direpresentasikan oleh rahasia matematis sebesar tiga puluh dua bita —dua ratus lima puluh enam bit. Sebuah angka yang sulit diingat dan lebih sulit lagi untuk dituliskan tanpa kesalahan. Industri kriptografi memecahkan masalah ini pada tahun 2013 dengan standar kecil dan elegan yang disebut BIP39: sebuah cara untuk merepresentasikan dua ratus lima puluh enam bit tersebut sebagai urutan dua puluh empat kata yang diambil dari daftar resmi dua ribu empat puluh delapan kata. Aritmetika di baliknya sangat pas; mereka yang ingin melihatnya secara detail dapat menemukannya di catatan samping.

Perhitungan dimulai dari akhir. Kita ingin merepresentasikan dua ratus lima puluh enam bit rahasia dengan menambahkan delapan bit checksum: total dua ratus enam puluh empat bit. Jika kita membaginya menjadi dua puluh empat kata —jumlah yang dapat dikelola untuk dicatat dan didektekan tanpa kehilangan data— setiap kata

harus memberikan tepat sebelas bit informasi. Dan sebelas bit adalah dua pangkat sebelas kemungkinan, yaitu dua ribu empat puluh delapan. Itulah sebabnya kosakata resmi BIP39 memiliki ukuran tepat sebesar itu: daftar tersebut ada sesuai dengan ukuran masalahnya, bukan sebaliknya.

Perhitungan tersebut bukan hiasan. Jika seseorang menuliskan dua puluh tiga kata dengan benar dan salah pada kata kedua puluh empat, checksum akan mendeteksinya: perangkat lunak akan memberi tahu mereka "urutan ini tidak valid". Jika seseorang menuliskan kedua puluh empat kata dengan benar, perangkat lunak akan menurunkan identitas yang sama tanpa ambiguitas. Pemilihan daftar kata juga disengaja: kata-kata dalam kosakata BIP39 pendek, berbeda satu sama lain, tanpa tanda diakritik, dipilih untuk meminimalkan kebingungan fonetik dan ejaan. Ini adalah kosakata yang dirancang untuk diingat, ditulis, dan didektekan oleh manusia tanpa kehilangan data.

Dari frasa ke kunci

Dua puluh empat kata tersebut bukanlah kunci kriptografi yang menandatangani pesan. Kata-kata itu adalah representasi yang dapat dipulihkan dari entropi asli yang, melalui proses deterministik yang disebut PBKDF2, diubah menjadi benih (seed) enam puluh empat byte. Dari benih itu diturunkan, juga secara deterministik, kunci kriptografi konkret yang digunakan pengguna: kunci pribadi untuk menandatangani dan kunci publik terkait yang dipublikasikan untuk memverifikasi tanda tangan. Mekanisme yang sama dalam sistem yang berbeda: mata uang kripto menggunakan kurva secp256k1; protokol Signal dan banyak sistem modern menggunakan Ed25519 pada kurva Curve25519. Untuk kurva konkret seperti Ed25519, standar BIP32 dan SLIP-0010 mengambil benih enam puluh empat byte itu dan menurunkan secara deterministik tiga puluh dua byte yang membentuk kunci tanda tangan efektif — tiga puluh dua byte yang sama dengan contoh kode di bagian selanjutnya dimulai.

Ini adalah cara standar di mana seluruh industri menyajikan mekanisme tersebut kepada pengguna —dompet mata uang kripto, pengelola identitas terdesentralisasi, Signal dalam bagian identitas persistennya, Solo2 di antaranya—: pengguna, dalam praktiknya, tidak pernah melihat benih atau kunci turunannya. Ia melihat dua puluh empat kata saat membuat identitasnya dan, secara opsional, mencatatnya di atas kertas. Kata-kata itu kemudian berpindah antar perangkatnya ketika ia ingin memigrasikan identitas: ia memasukkannya ke dalam aplikasi baru, aplikasi menurunkan benih yang sama, kunci yang sama, identitas yang sama. Ini adalah mekanisme yang portabel, solid secara kriptografi dan, dalam batas wajar, mudah diingat.

Cara menandatangani dengan kunci (sentuhan Zig)

Di Zig, setelah Anda memiliki benih tiga puluh dua byte yang diturunkan dari dua puluh empat kata, menandatangani pesan dengan Ed25519 hanya membutuhkan beberapa baris saja:

```
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);
```

Operasi penandatanganan menghasilkan enam puluh empat byte —disebut tanda tangan— yang hanya dapat dihasilkan dari kunci pribadi yang sesuai. Verifikasi bersifat publik: siapa pun dengan kunci publik dapat memeriksa apakah tanda tangan sesuai dengan pesan. Tanpa kunci pribadi, tidak ada yang bisa menghasilkan

tanda tangan yang valid untuk pesan tersebut; dengan kunci publik, semua orang dapat mendeteksi apakah tanda tangan itu valid. Asimetri inilah yang memungkinkan penandatangan untuk membuktikan kepemilikan tanpa membagikan rahasianya.

Contoh sebelumnya adalah versi manual minimal. Dalam kode Solo2 yang sebenarnya, rangkaian ini melewati dua file, satu di JavaScript yang berada di browser pengguna dan merekonstruksi entropi dari dua puluh empat kata, satu lagi di Zig di dalam pustaka *zcatcrypto* yang mengambil entropi tersebut dan menurunkan kunci kriptografi spesifik. Dimulai dari sisi 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 };
}
```

Tiga puluh dua byte entropi tersebut, bersama dengan tiga puluh dua byte lainnya yang diturunkan pada langkah yang sama, dikirim ke modul WebAssembly Zig yang menghasilkan kunci Ed25519 yang sebenarnya. Fungsi lengkapnya, dengan pembersihan memori akhirnya, muat dalam satu layar:

```
// 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;
}
```

Dua detail layak dicatat. Pertama: seed yang sama selalu menghasilkan pasangan kunci yang sama — hal inilah yang memungkinkan pemulihan identitas dengan memasukkan dua puluh empat kata ke dalam perangkat baru. Kedua: seed dihapus secara eksplisit dari memori pada baris terakhir. Setelah titik itu, bahkan fungsi itu sendiri tidak dapat merekonstruksi kunci; kata-kata pengguna akan menjadi satu-satunya sumber.

Bagi siapa pun yang ingin memeriksanya dengan angka kecil. Skema tanda tangan dapat dijelajahi sepenuhnya dengan angka yang cukup kecil untuk melakukan perhitungan secara manual. Siapa pun yang memilih untuk tidak masuk ke aritmatika dapat melewati blok ini tanpa kehilangan alur artikel; siapa pun yang ingin melihat mekanisme bekerja langkah demi langkah akan menemukannya di sini. **Aturan publik**, yang dapat dibaca siapa saja: sebuah bilangan prima $p = 23$ (dalam Ed25519 sebenarnya sekitar tujuh puluh tujuh digit; kami menggunakan dua puluh tiga agar perhitungan muat dalam satu halaman), sebuah basis $g = 2$ yang urutannya dalam grup ini adalah $q = 11$, dan konvensi bahwa semua aritmatika dengan g dilakukan *módulo p* dan semua eksponen direduksi *módulo q*. **Pilihan pribadi**, satu-satunya dan tidak pernah dibagikan: rahasia $x = 6$. Itulah identitasnya.

Langkah 1 — Bagian publik dari identitas. Ini dihitung sekali dan dipublikasikan secara terbuka.

$y = g^x \bmod p$

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

Bagian publik dari identitas adalah **18**. Siapa pun dapat mengambilnya dan menggunakannya untuk memverifikasi tanda tangan yang dibuat dengan identitas ini. Tidak ada seorang pun, dengan hanya mengamati angka 18, yang dapat memulihkan rahasia 6: itulah masalah logaritma diskrit yang akan kita bahas kembali di akhir.

Langkah 2 — Menandatangani pesan. Pemegang identitas ingin menandatangani pesan $m = 7$. Dia mulai dengan memilih nilai acak baru $k = 4$, yang akan digunakan sekali saja dan tidak akan pernah dibagikan (dalam Ed25519 sebenarnya, k diturunkan secara deterministik dari pesan dan rahasia untuk menghindari bahaya penggunaan ulang, tetapi peran yang dimainkannya persis seperti ini). Kemudian dia menghitung tiga angka:

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

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

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

Tanda tangannya adalah pasangan **(r, s) = (16, 10)**. Ini dikirim secara terbuka bersama dengan pesan. Siapa pun dapat membacanya. Catatan didaktik: dalam Ed25519 sebenarnya fungsi H adalah SHA-512, yang kuat secara kriptografi; di sini kita menggunakan penyederhanaan $e = (r + m) \bmod q$ agar pembaca dapat menelusuri langkah-langkahnya tanpa perlu menghitung hash. Struktur algoritmanya sama.

Langkah 3 — Memverifikasi tanda tangan. Verifikator memiliki bagian publik $y = 18$, pesan $m = 7$, dan tanda tangan $(r, s) = (16, 10)$. Dia merekonstruksi e dengan cara yang sama — $e = (16 + 7) \bmod 11 = 1$ — dan memeriksa apakah persamaan ini terpenuhi:

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

Hitung kedua sisi secara terpisah:

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

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

Kedua sisi memberikan hasil **12**. Tanda tangan valid. Siapa pun dengan bagian publik 18 dapat mencapai kesimpulan ini tanpa pernah tahu bahwa rahasianya adalah 6.

Bagaimana dengan pihak ketiga yang mencoba memalsukan? Eva telah melihat semua yang publik melewati saluran: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Untuk menandatangani pesan yang *berbeda* atas nama identitas ini, dia perlu mengetahui x. Satu-satunya caranya adalah dengan bertanya pada diri sendiri: "untuk eksponen x berapa $2^x \bmod 23 = 18$ terpenuhi?". Dengan $p = 23$ dia dapat mencoba 0, 1, 2, 3, … dan menemukannya dalam hitungan detik. Tetapi saat mengganti 23 dengan bilangan prima dari dimensi sebenarnya dari Ed25519, ruang eksponen yang mungkin melebihi jumlah atom di alam semesta yang teramati. **Saat ini tidak ada algoritma yang dikenal manusia yang dapat menelusuri ruang tersebut dalam waktu kurang dari miliaran tahun.** Ini adalah masalah logaritma diskrit yang sama yang mendasari Diffie-Hellman dari artikel sebelumnya, yang diterapkan di sini pada skema tanda tangan.

Apa yang baru saja kita lalui adalah *persis* Schnorr, skema tanda tangan di mana Ed25519 adalah varian yang diadaptasi ke kurva elips. Dalam Ed25519 sebenarnya, semua operasi dilakukan pada titik-titik kurva tertentu (Curve25519) alih-alih pada bilangan bulat modulo bilangan prima, dan fungsi H adalah SHA-512 alih-alih penjumlahan mainan yang kita gunakan di atas. Kedua substitusi tersebut adalah penyesuaian implementasi — mendapatkan ketahanan kriptografi terhadap brute force, mendapatkan properti keamanan tambahan untuk k. Struktur algoritma, tiga operasi, alasan asimetri, adalah sama.

Sangat tepat untuk berhenti sejenak di sini, karena seluruh rangkaian dapat disalahartikan dalam pandangan sekilas dengan primitif lain dari ketiganya: hash. Ini bukan itu. Hash adalah fungsi unik yang mengompresi — banyak byte masuk, jejak pendek keluar, di situlah jalannya berakhir. Identitas kriptografi adalah pasangan matematika yang saling melengkapi: rahasia tetap ada dan menandatangani; rekan publiknya dipublikasikan dan memverifikasi. Di mana hash meruntuhkan informasi dalam satu arah, identitas membangun asimetri antara dua bagian. Hash membuktikan apa yang dikatakan; identitas membuktikan siapa yang mengatakannya.

Apa yang bukan merupakan frasa

Tiga kesalahpahaman umum perlu diluruskan. Frasa bukanlah kata sandi dalam arti yang sebenarnya: ia tidak dibandingkan dengan sidik jari yang disimpan di server; ia dimasukkan ke dalam perangkat pengguna untuk merekonstruksi identitas secara matematis. Frasa tidak dapat dipulihkan: jika hilang, tidak ada orang yang bisa dimintai bantuan; jika digandakan, identitasnya juga ikut digandakan. Frasa bukanlah kredensial yang dapat dipisahkan dari identitas: frasa *adalah* identitas. Siapa pun yang memilikinya dapat bertindak atas nama identitas tersebut, tanpa izin tambahan, tanpa proses otorisasi, tanpa kemungkinan pemulihan.

Properti ketiga inilah yang mengubah bobot masalahnya. Kata sandi yang hilang adalah gangguan administratif. Identitas kriptografi yang hilang adalah identitas itu sendiri. Kertas berisi frasa yang ditemukan oleh pihak ketiga bukanlah risiko pencurian akun: melainkan penyerahan seluruh identitas. Janji sistem —bahwa tidak ada yang dapat mencabut identitas Anda atau memblokir Anda secara sewenang-wenang— disertai secara tak terpisahkan dengan tanggung jawab —bahwa Anda adalah satu-satunya penjaga sesuatu yang tidak dapat dipulihkan oleh siapa pun untuk Anda.

Janji dan bobot

Model identitas kriptografi biasanya disebut sebagai *berdaulat sendiri* —self-sovereign dalam literatur bahasa Inggris—. Pemilihan kata ini disengaja dan menggambarkan kondisinya dengan cukup akurat. Pengguna berdaulat atas identitasnya dalam arti yang hampir mirip dengan abad pertengahan: tidak ada raja, penerbit, atau otoritas pusat yang memberikannya; tidak satu pun dari mereka dapat menariknya kembali. Namun, seperti raja abad pertengahan, pengguna memikul seluruh konsekuensi dari kesalahannya: tidak ada bupati yang akan mengambil keputusan menggantikannya jika ia kehilangan segelnya.

Pilihan antara identitas yang dikelola oleh pihak ketiga dan identitas berdaulat sendiri tidak memiliki jawaban benar yang universal. Untuk akun forum yang tidak penting, identitas yang dikelola mungkin sebanding dengan risikonya. Namun untuk identitas profesional yang menandatangani dokumen yang mengikat secara hukum, untuk identitas ekonomi yang menjaga tabungan sendiri, untuk identitas komunikasi profesional dengan klien yang telah mempercayakan informasi sensitif, masalahnya berubah. Di sana pertanyaannya bukan lagi «apakah nyaman?» dan menjadi «siapa, selain saya, yang memiliki kekuasaan untuk bertindak atas nama saya, dan dalam keadaan apa?».

Di mana mekanisme ini muncul dalam sistem nyata

BIP39 lahir di dunia Bitcoin pada tahun 2013 dan menyebar dengan cepat ke seluruh ekosistem mata uang kripto: dompet serius mana pun saat ini menerima frasa BIP39 yang terdiri dari dua belas atau dua puluh empat kata sebagai cadangan identitas ekonomi pemiliknya. Di luar mata uang kripto, konsep dasar yang sama — pasangan kriptografi yang membuktikan kepengarangan tanpa perantara — muncul di sistem lain dengan sintaksis yang berbeda. Kunci SSH yang digunakan administrator sistem untuk mengakses server mereka adalah kasus klasik: kunci privat yang disimpan administrator di mesin mereka dan kunci publik yang disalin ke setiap server; tidak ada entitas yang sebanding dengan layanan terpusat yang mengintervensi. Protokol Signal menggunakan Ed25519 dengan materi kunci persisten pada perangkat; eIDAS Eropa, dalam bagian tanda tangan terkualifikasi, bertumpu pada prinsip kriptografi yang sama, dengan perbedaan bahwa kunci disimpan oleh penyedia layanan kepercayaan terkualifikasi, bukan oleh pengguna.

Solo2, platform penerbit publikasi ini, menggunakan frasa BIP39 dua puluh empat kata sebagai identitas setiap pengguna. Pengguna, saat membuat akun, melihat kata-kata tersebut sekali. Kata-kata itu tidak disimpan di server Solo2 mana pun atau milik siapa pun: jika pengguna mencatat dan menjaganya, mereka mempertahankan identitas mereka selamanya. Jika mereka kehilangannya, mereka kehilangannya. Ini adalah konsekuensi logis dari arsitektur tanpa operator di tengah: jika Solo2 dapat mengembalikan identitas kepada pengguna yang kehilangannya, ia juga dapat memberikannya kepada siapa pun yang menekan Solo2 untuk memberikannya.

Untuk pembaca profesional

Empat pertimbangan bagi mereka yang mengevaluasi penerapan identitas kriptografi kedaulatan mandiri (autosoberana) dalam konteks profesional:

1. Frasa tersebut adalah identitas. Penjagaan fisik — kertas, beberapa salinan di tempat berbeda, akhirnya logam yang digrafir untuk penggunaan jangka panjang — menawarkan lebih banyak jaminan daripada penjagaan digital, yang menambah permukaan serangan tanpa mengurangi risiko kehilangan.
2. Tidak ada pemulihan. Merancang proses dengan asumsi bahwa suatu hari salinan utama akan hilang jauh lebih bijaksana daripada menemukannya di hari saat salinan tersebut hilang. Salinan kedua yang terpisah secara geografis menyelesaikan hampir semua skenario.
3. Ini tidak sama dengan sertifikat terkualifikasi eIDAS. Untuk tanda tangan terkualifikasi di Uni Eropa — akta notaris, prosedur tertentu dengan Administrasi — undang-undang mewajibkan penyedia terkualifikasi yang menyimpan kuncinya. Identitas kriptografi kedaulatan mandiri berfungsi untuk komunikasi profesional dan penandatanganan dokumen dengan nilai pembuktian, tetapi tidak secara otomatis menggantikan sertifikat terkualifikasi dalam kasus di mana norma mewajibkannya.
4. Jika identitas akan ditransfer — warisan, suksesi profesional, penutupan aktivitas — disarankan untuk mempersiapkan prosedurnya sebelum, bukan sesudah. Prosedur formal dengan amplop yang disegel lilin

(lacre), instruksi kepada pelaksana wasiat, penitipan di kantor notaris, adalah pengaturan klasik yang sangat kompatibel dengan sifat kriptografi aset tersebut.

Artikel ini menutup trio konseptual yang membuka siklus — hash, enkripsi, identitas —. Ketiga ide tersebut dibangun satu sama lain: hash memberikan sidik jari yang tidak dapat diubah, enkripsi memberikan kerahasiaan tanpa pihak ketiga yang tepercaya, identitas memberikan kepengarangan tanpa pihak ketiga yang memberikan. Ketiganya berbagi properti yang juga bukan ideologis: mereka mentransfer, dari pihak yang mengelola layanan ke pihak yang menggunakannya, kemampuan teknis yang secara tradisional berada di tangan operator. Mereka juga mentransfer tanggung jawab bersamanya. Berbicara jujur tentang salah satu dari ketiganya mengharuskan kita untuk juga berbicara tentang dua lainnya.

Sumber dan bacaan lebih lanjut

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, proposal peningkatan Bitcoin tahun 2013. Standar de facto untuk frasa pemulihan dalam industri kripto.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), termasuk Ed25519. IETF, Januari 2017. Spesifikasi normatif dari skema tanda tangan yang digunakan di sebagian besar industri kontemporer.
- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, versi 2.0. IETF, September 2000. Mendefinisikan algoritma PBKDF2 yang digunakan dalam derivasi BIP39 dari frasa ke seed.
- Regulasi (EU) 910/2014 (eIDAS) dan evolusinya oleh Regulasi (EU) 2024/1183 (eIDAS 2) — kerangka kerja Eropa untuk identitas elektronik dan tanda tangan terkualifikasi. Rezim yang berbeda dari kedaulatan mandiri, tetapi secara konseptual didukung oleh primitif kriptografi yang sama.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Teks kanonik tentang prinsip dan komitmen model kedaulatan mandiri, sebelumnya tetapi relevan untuk memahami keluarga solusi kontemporer.

← SebelumnyaModel bisnis sebagai sinyal kepercayaanBerikutnya →Self-hosting sebagai praktik profesional

Bacaan terbaru

- Refleksi · 29 Juni 2026 Kamu tidak anonim
- Refleksi · 27 Mei 2026 Apa yang tidak bisa diperbaiki oleh tanda tangan
- Analisis · 26 Mei 2026 Privasi nyata vs semu: pertanyaan yang perlu Anda ajukan

Bawa artikel ini bersama Anda ke mana pun Anda membutuhkannya.

↓ Markdown ↓ Teks murni ↓ PDF

File akan diunduh ke perangkat Anda. Dari sana Anda dapat menyimpannya, mengimpornya ke Solo2, atau membagikannya di mana pun Anda mau. Cuadernos tidak memutuskan tujuan untuk Anda.

Segel lilin · SHA-256 3d47a8f7f93be314f7d607a504c45a316a75d8044c6c63e66056f92fe3ec6860

Cuadernos Lacre · Publikasi dari Menzuri Gestión S.L. ·
ditulis oleh R.Eugenio · disunting oleh tim Solo2.

perangkat Anda sendiri. Tanpa sumber daya pihak ketiga, tanpa pelacak, tanpa pemrofilan, tanpa berbagi data. Jika Anda ingin mengikuti kami: RSS.