24 slov: co je to kryptografická identita

Kryptografická identita není heslo: neukládá ji žádný server a nelze ji obnovit. Didaktické vysvětlení mechanismu BIP39, proč přesně čtyřiadvacet slov a jaká reálná váha spočívá na tom, kdo je vlastní.

Abychom si rozuměli: Pokud zapomenete heslo ke Gmailu, Google vám ho resetuje. Pokud ztratíte 24 slov, která tvoří kryptografickou identitu, nemáte koho požádat o jejich vrácení. Nejde o to, že by byl postup přísný — jde o to, že na druhém konci nikdo není. Tento rozdíl je naprosto zásadní.

Rozdíl mezi heslem a identitou

Heslo v klasickém modelu internetu není identitou uživatele. Je to doklad. Uživatel má identitu — jméno, e-mail, zákaznické číslo — a aby serveru dokázal, že je tím, za koho se vydává, předloží heslo, které server porovná s otiskem, jejž má uložený. Pokud se otisky shodují, server povolí relaci. Pokud se heslo ztratí, uživatel zůstává stejným uživatelem; to, co ztrácí, je doklad, a existuje postup pro obnovu — e-mail na registrovanou adresu, bezpečnostní otázka — jak jej získat zpět.

Kryptografická identita funguje jinak. Není to pověření, které by někdo porovnával s uloženým otiskem; *je to* úplné matematické tajemství samo o sobě. Je jedno, kde se nachází — na papíře, v zařízení, nebo dokonce na cizím serveru — identita existuje díky své matematice, nikoli díky tomu, kdo ji ověřuje. Zde se objevuje vlastnost podobná té, kterou jsme viděli v článku «Co je vlastně SHA-256»: vlastnictví se nedokazuje předložením tajemství, ale jeho použitím k podpisu. Takto vytvořený podpis může kdokoli ověřit pomocí veřejné hodnoty, která je matematicky odvozena ze samotného tajemství, aniž by tajemství musel znát a aniž by do ověřování zasahovala třetí strana. Kdo má tajemství, je identitou; kdo ho ztratí, přestává jí být. Rozsudek je kategorický: **neexistuje nikdo, koho byste mohli požádat o vrácení identity. Takový někdo neexistuje, protože ji v první řadě vůbec neměl.**

Co představuje čtyřiadvacet slov

Kryptografická identita je obvykle reprezentována matematickým tajemstvím o délce dvaatřiceti bajtů — dvě stě padesáti šesti bitů. Číslo, které je těžké si zapamatovat a ještě těžší ho bezchybně opsat. Kryptografický průmysl vyřešil tento problém v roce 2013 malým a elegantním standardem zvaným BIP39: způsobem, jak reprezentovat těchto dvě stě padesát šest bitů jako sekvenci čtyřiadvaceti slov vybraných z oficiálního seznamu dvou tisíc čtyřiceti osmi slov. Aritmetika v pozadí do sebe elegantně zapadá; kdo ji chce vidět podrobně, najde ji v poznámce na okraji.

Počítání začíná od konce. Chceme reprezentovat dvě stě padesát šest bitů tajemství a přidat osm bitů kontrolního součtu: celkem dvě stě šedesát čtyři bitů. Pokud je rozdělíme do čtyřiadvaceti slov — což je zvládnutelný počet pro zápis i diktování beze ztrát — musí každé slovo nést přesně jedenáct bitů informace. A jedenáct bitů je dvě na jedenáctou možností, tedy dva tisíce čtyřicet osm. Proto má oficiální slovník BIP39 právě tuto velikost: seznam existuje na míru problému, nikoli naopak.

Počítání není dekorativní. Pokud někdo opíše třiadvacet slov správně a ve čtyřiadvacátém se splete, kontrolní součet to zjistí: software mu řekne «tato sekvence není platná». Pokud někdo opíše všech čtyřiadvacet slov

správně, software jednoznačně odvodí stejnou identitu. Volba seznamu slov je také záměrná: slova ze slovníku BIP39 jsou krátká, vzájemně odlišná, bez diakritiky, zvolená tak, aby se minimalizovaly fonetické a pravopisné záměny. Je to slovník navržený tak, aby si ho lidé zapamatovali, zapsali a nadiktovali beze ztrát.

Od fráze ke klíči

Těch čtyřiadvacet slov není kryptografický klíč, kterým se podepisují zprávy. Jsou obnovitelnou reprezentací původní entropie, která se pomocí deterministického procesu zvaného PBKDF2 transformuje na šedesátibytový seed. Z tohoto seedu se rovněž deterministicky odvozují konkrétní kryptografické klíče, které uživatel používá: soukromý klíč k podepisování a odpovídající veřejný klíč, který se zveřejňuje pro ověřování podpisů. Stejný mechanismus v různých systémech: kryptoměny používají křivku secp256k1; protokol Signal a mnoho moderních systémů používají Ed25519 na křivce Curve25519. Pro konkrétní křivku, jako je Ed25519, berou standardy BIP32 a SLIP-0010 onen šedesátibytový seed a deterministicky odvozují dvaatřicet bajtů, které tvoří efektivní podpisový klíč — stejných dvaatřicet bajtů, kterými začíná příklad kódu v následující části.

Toto je standardní způsob, jakým celý průmysl prezentuje mechanismus uživateli —kryptoměnové peněženky, správci decentralizované identity, Signal ve své části pro trvalou identitu, Solo2 mezi nimi—: uživatel v praxi nikdy nevidí seed ani odvozené klíče. Vidí oněch čtyřiadvacet slov při vytváření své identity a volitelně si je zapíše na papír. Slova pak cestují mezi jeho zařízeními, když chce identitu migrovat: zadá je do nové aplikace, aplikace odvodí stejný seed, stejné klíče, stejnou identitu. Je to přenosný, kryptograficky solidní a v mezích rozumného zapamatovatelný mechanismus.

Jak se podepisuje klíčem (nástin v Zig)

V Zig, jakmile máte dvaatřicetibytový seed odvozený z čtyřiadvaceti slov, podepsání zprávy pomocí Ed25519 se vejde na pár řádků:

```
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);
```

Operace podepisování vytvoří šedesát čtyři bajtů —nazývaných podpis— které mohly být vygenerovány pouze z odpovídajícího soukromého klíče. Ověření je veřejné: kdokoli s veřejným klíčem může zkontrolovat, zda podpis odpovídá zprávě. Bez soukromého klíče nemůže nikdo vytvořit platný podpis pro danou zprávu; s veřejným klíčem může kdokoli zjistit, zda je podpis platný. Tato asymetrie je to, co umožňuje podepisujícímu prokázat autorství bez sdílení tajemství.

Předchozí příklad je minimální verze manuálu. V reálném kódu Solo2 řetězec prochází dvěma soubory, jedním v JavaScriptu, který běží v prohlížeči uživatele a rekonstruuje entropii z čtyřiadvaceti slov, a druhým v Zig v knihovně *zcatcrypto*, který tuto entropii přebírá a odvozuje konkrétní kryptografické klíče. Počínaje stranou prohlížeče:

```
// 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 };
}
```

Těchto dvaatřicet bajtů entropie spolu s dalšími dvaatřiceti odvozenými ve stejném kroku putuje do modulu WebAssembly v Zig, který generuje samotné klíče Ed25519. Kompletní funkce s konečným vyčištěním paměti se vejde na jednu obrazovku:

```
// 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;
}
```

Za zmínku stojí dva detaily. První: stejný seed produkuje vždy stejný pár klíčů — právě to umožňuje obnovit identitu zadáním čtyřiadvaceti slov do nového zařízení. Druhý: seed se v posledním řádku explicitně vymaže z paměti. Po tomto bodě by ani samotná funkce neměla být schopna klíče rekonstruovat; slova uživatele by byla jediným zdrojem.

Pro ty, kteří si to chtějí ověřit na malých číslech. Schéma podpisu lze projít celé s čísly dostatečně malými na to, aby se výpočty daly provádět ručně. Kdo raději nechce zabíhat do aritmetiky, může tento blok přeskočit, aniž by ztratil nit článku; kdo chce vidět mechanismus fungující krok za krokem, najde ho zde. **Veřejná pravidla**, která si může přečíst kdokoli: prvočíslo $p = 23$ (v reálném Ed25519 má asi sedmdesát sedm číslic; používáme třiadvacet, aby se výpočty vešly na jednu stránku), základ $g = 2$, jehož řád v této grupě je $q = 11$, a konvence, že veškerá aritmetika s g se provádí *módulo p* a všechny exponenty se redukují *módulo q*. **Soukromá volba**, jediná a nikdy nesdílená: tajemství $x = 6$. To je identita.

Krok 1 — Veřejná část identity. Vypočítá se jednou a otevřeně se zveřejní.

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

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

Veřejná část identity je **18**. Kdokoli ji může vzít a použít k ověření podpisů vytvořených s touto identitou. Nikdo, kdo pozoruje pouze 18, nemůže obnovit tajemství 6: to je problém diskrétního logaritmu, ke kterému se vrátíme na konci.

Krok 2 — Podepsání zprávy. Držitel identity chce podepsat zprávu $m = 7$. Začne výběrem nové náhodné hodnoty $k = 4$, která se použije pouze jednou a nikdy nebude sdílena (v reálném Ed25519 se k odvozuje deterministicky ze zprávy a tajemství, aby se předešlo nebezpečí jeho opětovného použití, ale role, kterou hraje, je přesně tato). Poté vypočítá tři čísla:

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

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

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

Podpis je dvojice **(r, s) = (16, 10)**. Putuje otevřeně spolu se zprávou. Kdokoli jej může přečíst. Didaktická poznámka: v reálném Ed25519 je funkce H SHA-512, kryptograficky robustní; zde používáme zjednodušení $e = (r + m) \bmod q$, aby si čtenář mohl projít kroky bez nutnosti výpočtu hashe. Struktura algoritmu je stejná.

Krok 3 — Ověření podpisu. Ověřovatel má veřejnou část $y = 18$, zprávu $m = 7$ a podpis $(r, s) = (16, 10)$. Rekonstruuje e stejným způsobem —$e = (16 + 7) \bmod 11 = 1$— a zkontroluje, zda tato rovnost platí:

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

Vypočítá obě strany zvlášť:

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

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

Obě strany dávají **12**. Podpis je platný. Kdokoli s veřejnou částí 18 může dospět k tomuto závěru, aniž by kdy věděl, že tajemství bylo 6.

A co třetí strana, která by se pokusila o padělání? Eva viděla vše veřejné procházející kanálem: $p = 23$, $g = 2$, $q = 11$, $y = 18$, $m = 7$, $r = 16$, $s = 10$. Aby mohla podepsat *jinou* zprávu jménem této identity, musela by znát x. Její jedinou cestou je zeptat se sama sebe: „pro jaký exponent x platí $2^x \bmod 23 = 18$?“. S $p = 23$ může zkusit 0, 1, 2, 3, … a najít ho během několika sekund. Ale při nahrazení 23 prvočíslem reálných rozměrů Ed25519 prostor

možných exponentů přesahuje počet atomů v pozorovatelném vesmípy. **Dodnes neexistuje žádný lidstvu známý algoritmus, který by dokázal projít tento prostor za méně než miliardy let.** Je to stejný problém diskrétního logaritmu, na kterém staví Diffie-Hellman v předchozím článku, aplikovaný zde na schéma podpisu.

To, co jsme právě prošli, je *přesně* Schnorr, schéma podpisu, jehož je Ed25519 variantou přizpůsobenou eliptické křivce. V reálném Ed25519 se všechny operace provádějí nad body konkrétní křivky (Curve25519) namísto celých čísel modulo prvočíslo a funkce *H* je SHA-512 namísto námi použitého zjednodušeného součtu. Obě substituce jsou úpravy implementace — získání kryptografické odolnosti proti hrubé síle, získání dalších bezpečnostních vlastností pro *k*. Algoritmická struktura, tři operace a důvod asymetrie jsou stejné.

Zde je vhodné se krátce zastavit, protože celý řetězec si lze při zběžném pohledu splést s jiným primitivem z trojice: hashem. Není jím. Hash je unikátní funkce, která komprimuje — vstupuje mnoho bajtů, vystupuje krátký otisk a tam cesta končí. Kryptografická identita je matematicky doplňková dvojice: tajemství zůstává a podepisuje; jeho veřejný protějšek se zveřejňuje a ověřuje. Zatímco hash kolabuje informace v jednom směru, identita vytváří asymetrii mezi dvěma polovinami. Hash potvrzuje, co bylo řečeno; identita potvrzuje, kdo to řekl.

Čím fráze není

Je vhodné vyjasnit tři časté omyly. Fráze není heslo v pravém slova smyslu: neporovnává se s otiskem uloženým na serveru; zadává se do zařízení uživatele za účelem matematické rekonstrukce identity. Fráze se neobnovuje: pokud se ztratí, není nikdo, koho byste o ni mohli požádat; pokud se duplikuje, duplikuje se i identita. Fráze není pověření oddělitelné od identity: fráze *je* identita. Kdo ji má, může za ni jednat, bez dalšího svolení, bez procesu autorizace, bez možnosti obnovy.

Právě tato třetí vlastnost mění váhu celé věci. Ztracené heslo je administrativní nepříjemnost. Ztracená kryptografická identita je identita sama. Papír s frází nalezený třetí stranou není riziko krádeže účtu: je to odevzdání celé identity. Příslib systému — aby vám nikdo nemohl identitu odebrat nebo vás svévolně zablokovat — je neoddělitelně doprovázen odpovědností — že vy jste jediným strážcem něčeho, co za vás nikdo nemůže obnovit.

Příslib a váha

Model kryptografické identity se obvykle označuje jako *samosuverénní* —self-sovereign v anglosaské literatuře —. Volba slova je záměrná a popisuje stav dosti přesně. Uživatel je suverénem nad svou identitou v téměř středověkém smyslu: neuděluje ji žádný král, žádný vydavatel, žádná centrální autorita; ani ji žádný z výše uvedených nemůže odebrat. Ale stejně jako středověký monarcha nese uživatel veškeré následky svých chyb: neexistuje žádný regent, který by rozhodoval za něj, pokud ztratí pečeť.

Volba mezi identitou spravovanou třetí stranou a samosuverénní identitou nemá jedinou univerzálně správnou odpověď. Pro účet na nepodstatném fóru je spravovaná identita pravděpodobně úměrná riziku. Pro profesní identitu, která podepisuje právně závazné dokumenty, pro ekonomickou identitu, která střeží vlastní úspory, pro identitu profesní komunikace s klienty, kteří svěřili citlivé informace, se situace mění. Tam otázka přestává být „je to pohodlné?“ a stává se „kdo kromě mě má moc jednat mým jménem a za jakých okolností?“.

Kde se tento mechanismus objevuje v reálných systémech

BIP39 se zrodil ve světě Bitcoin v roce 2013 a rychle se rozšířil do celého kryptoměnového ekosystému: každá seriózní peněženka dnes přijímá dvanácti- nebo čtyřiadvacetislovnou frázi BIP39 jako zálohu ekonomické identity svého držitele. Mimo kryptoměny se stejný základní koncept — kryptografický pár prokazující autorství bez prostředníka — objevuje v jiných systémech s odlišnou syntaxí. SSH klíče, které správce systému používá k přístupu ke svým serverům, jsou klasickým případem: soukromý klíč, který si správce ukládá na svém stroji, a

veřejný, který se kopíruje na každý server; nezasahuje žádný subjekt srovnatelný s centralizovanou službou. Protokol Signal používá Ed25519 s perzistentním materiálem klíče v zařízení; evropské eIDAS se ve své části o kvalifikovaném podpisu opírají o stejný kryptografický princip s tím rozdílem, že klíč opatruje kvalifikovaný poskytovatel důvěryhodných služeb namísto uživatele.

Solo2, vydavatelská platforma této publikace, používá čtyřiadvacetislovnou frázi BIP39 jako identitu každého uživatele. Uživatel při vytváření svého účtu vidí slova jednou. Neukládají se na žádném serveru Solo2 ani nikoho jiného: pokud si je uživatel poznamená a opatruje, uchová si svou identitu navždy. Pokud je ztratí, ztratí je. Je to logický důsledek architektury bez operátora uprostřed: kdyby Solo2 mohla vrátit identitu uživateli, který ji ztratil, mohla by ji také dát komukoli, kdo na Solo2 zatlačí, aby ji vydala.

Pro profesionálního čtenáře

Čtyři úvahy pro ty, kteří zvažují přijetí kryptografické samovrstevné (autosoberana) identity v profesionálním kontextu:

1. Fráze je identita. Fyzické opatrování — papír, několik kopií na různých místech, případně kov s gravírováním pro dlouhodobé použití — nabízí více záruk než digitální opatrování, které zvětšuje útočnou plochu, aniž by snižovalo riziko ztráty.
2. Neexistuje žádná obnova. Navrhnout proces za předpokladu, že jednoho dne dojde ke ztrátě primární kopie, je mnohem vhodnější než to zjistit v den, kdy ke ztrátě dojde. Druhá geograficky oddělená kopie řeší téměř všechny scénáře.
3. Není to totéž co kvalifikovaný certifikát eIDAS. Pro kvalifikovaný podpis v Unii — notářské zápisy, určité úkony s úřady — legislativa vyžaduje kvalifikovaného poskytovatele, který klíč opatruje. Kryptografická samovrstevná identita slouží pro profesionální komunikaci a podepisování dokumentů s důkazní hodnotou, ale nenahrazuje automaticky kvalifikovaný certifikát v případech, kdy to norma vyžaduje.
4. Pokud má být identita převedena — dědictví, profesní nástupnictví, ukončení činnosti — je vhodné připravit postup předem, nikoli až poté. Formální postupy s obálkami zapečetěnými pečetním voskem (lacre), instrukce pro vykonavatele závěti, uložení u notáře, jsou klasická ujednání dokonale slučitelná s kryptografickou povahou aktiva.

Tento článek uzavírá konceptuální trio, které cyklus otevřelo — hash, šifrování, identita —. Tyto tři myšlenky se staví jedna na druhé: hash dává neměnný otisk, šifrování dává důvěrnost bez důvěryhodné třetí strany, identita dává autorství bez poskytující třetí strany. Všechny tři sdílejí vlastnost, která také není ideologická: přenášejí od toho, kdo spravuje službu, na toho, kdo ji používá, technické možnosti, které tradičně spočívaly na operátorovi. Spolu s nimi přenášejí i odpovědnost. Mluvit čestně o kterékoli z těchto tří vyžaduje mluvit i o zbývajících dvou.

Zdroje a další čtení

- Palatinus, M.; Rusnak, P.; Voisine, A.; Bowe, S. — *BIP-0039: Mnemonic code for generating deterministic keys*, návrh na vylepšení Bitcoin z roku 2013. De facto standard pro fráze pro obnovu v kryptoprůmyslu.
- RFC 8032 — Edwards-Curve Digital Signature Algorithm (EdDSA), včetně Ed25519. IETF, leden 2017. Normativní specifikace schématu podpisu používaného ve velké části současného průmyslu.
- RFC 2898 — PKCS #5: Password-Based Cryptography Specification, verze 2.0. IETF, září 2000. Definuje algoritmus PBKDF2 použitý v derivaci BIP39 z fráze na seed.
- Nařízení (EU) 910/2014 (eIDAS) a jeho vývoj nařízením (EU) 2024/1183 (eIDAS 2) — evropský rámec pro elektronickou identitu a kvalifikovaný podpis. Jiný režim než samovrstevný, ale konceptuálně se opírající o stejná kryptografická primitiva.
- Allen, C. — *The Path to Self-Sovereign Identity* (2016). Kanonický text o principech a závazcích samovrstevného modelu, starší, ale relevantní pro pochopení rodiny současných řešení.

← PředchozíObchodní model jako signál důvěryDalší →Self-hosting jako profesionální praxe

Nedávné čtení

- Zamyšlení · 29. června 2026 Nejste anonymní
- Zamyšlení · 27. května 2026 Co podpis nemůže napravit
- Analýza · 26. května 2026 Skutečné vs. zdánlivé soukromí: otázky, které je vhodné si položit

Vezměte si tento článek tam, kam potřebujete.

↓ Markdown ↓ Prostý text ↓ PDF

Soubor se stáhne do vašeho zařízení. Odtud si jej můžete uložit, importovat do Solo2 nebo sdílet, kdekoli chcete. Cuadernos za vás o cíli nerozhoduje.

Pečeť · SHA-256 7638d929ad9661bbd4842ba4baf2a15a0842ba355c174aca380bf7c543aa548d

Cuadernos Lacre · Publikace Menzuri Gestión S.L. ·
napsal R.Eugenio · rediguje tým Solo2.