Bezpieczeństwo Palladin

Co chroni Palladin i gdzie przebiegają granice zaufania.

1.0 25.10.2026

Model bezpieczeństwa

Jak Palladin chroni dane?

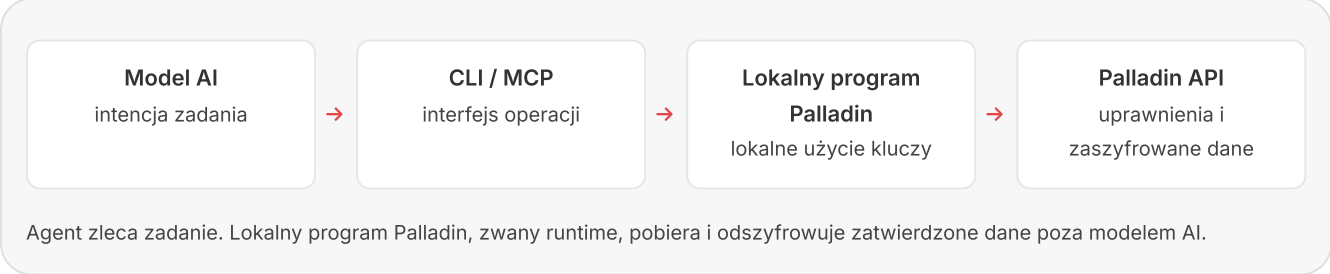
Palladin pozwala agentom AI korzystać z haseł i kluczy API pod Twoją kontrolą. Dane są szyfrowane na Twoim urządzeniu. Ty decydujesz, do czego agent ma dostęp i jak może użyć tych danych.

Gwarancja zero-knowledge

Zero-knowledge oznacza, że serwer przechowuje zaszyfrowaną zawartość sejfu bez kluczy potrzebnych do jej odczytania. Szyfrowanie i odszyfrowanie odbywa się w aplikacji użytkownika lub lokalnym programie agenta.

Serwer nadal obsługuje konta, uprawnienia i informacje o operacjach. Klucze API używane do połączenia z Palladin to osobna kategoria od sekretów zapisanych w sejfie.

Kto ma dostęp do danych?



Agent zleca zadanie. Lokalny program Palladin, zwany runtime, pobiera i odszyfrowuje zatwierdzone dane poza modelem AI.

Uczestnik	Odpowiedzialność
Klient użytkownika	Odblokowanie, szyfrowanie i zatwierdzenie dostępu.
Backend	Tożsamość, aktualny stan uprawnień, limity i przechowywanie zaszyfrowanych danych.
Lokalny program Palladin	Sprawdzenie danych, ich odszyfrowanie i wykonanie dozwolonej operacji.
Model AI	Zlecenie zadania; dostęp do odszyfrowanych danych zależy od wybranej metody.
Proces lub strona docelowa	Otrzymuje hasło lub klucz potrzebny do zadania. Musisz mu ufać.

Trzy rodzaje kontroli

Szyfrowanie chroni treść. **Serwer** sprawdza uprawnienia. **Lokalny program Palladin** wykonuje dozwoloną operację. Każda warstwa ma inne zadanie. Zablokowanie kolejnego pobrania nie usuwa danych, które odbiorca wcześniej skopiował.

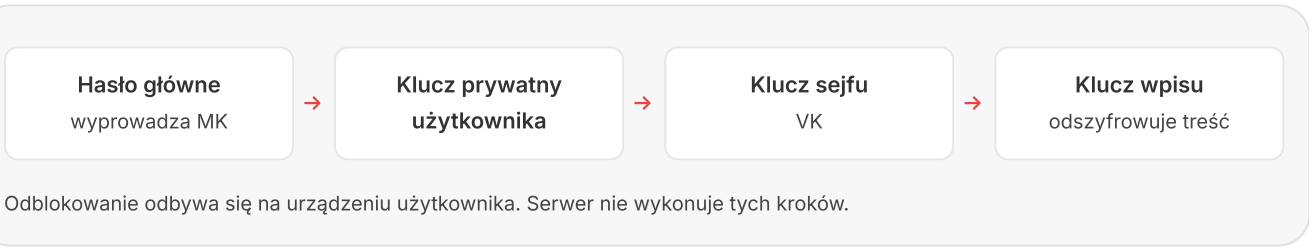
Dane i klucze

Po co kilka kluczy?

Palladin używa osobnych kluczy dla użytkownika, sejfu i wpisu. Dzięki temu udostępnienie jednego wpisu nie musi oznaczać przekazania klucza do całego sejfu.

MK — Master Key, czyli klucz główny — powstaje z hasła głównego przy użyciu Argon2id. To algorytm, który utrudnia masowe zgadywanie hasła. MK chroni klucz prywatny użytkownika.

VK — Vault Key, czyli klucz sejfu — pozwala otworzyć klucze wpisów w danym sejfie. Każdy uprawniony członek otrzymuje własną zaszyfrowaną kopię VK, którą może otworzyć swoim kluczem prywatnym. Taką zabezpieczoną kopię nazywamy **kopertą klucza**.



Odblokowanie odbywa się na urządzeniu użytkownika. Serwer nie wykonuje tych kroków.

Pozostałe klucze

Nazwa w protokole	Co chroni
EntryDEK — klucz szyfrowania wpisu	Treść i opis wpisu dla członków sejfu. DEK oznacza Data Encryption Key, czyli klucz szyfrowania danych.
GrantDEK — klucz przyznanego dostępu	Kopię danych przygotowaną dla konkretnej zgody na dostęp do wpisu.
VDK — klucz wyszukiwania w sejfie	Metadane, które agent może wyszukiwać, bez dostępu do wartości sekretu.
Klucze agenta X25519 i Ed25519	Odpowiednio: odbiór zaszyfrowanych danych i podpisywanie żądań agenta.

Co widzi serwer?

Serwer przechowuje zaszyfrowane dane i koperty kluczy. Zna identyfikatory organizacji, sejfów, wpisów i agentów oraz ich uprawnienia, wersje i zdarzenia audytu. Szyfrowanie nie ukrywa całego faktu korzystania z usługi.

Dane do wyszukiwania są szyfrowane oddzielnie od sekretów. Agent może więc znaleźć wpis bez otrzymania jego hasła lub klucza API.

Ochrona przed podmianą

Przed odszyfrowaniem aplikacja sprawdza, czy dane należą do właściwego sejfu, odbiorcy i przyznanego dostępu oraz czy mają właściwą wersję. Porównuje je z zaufanym stanem połączenia i uprawnień, a nie tylko z opisem dołączonym do otrzymanej paczki.

Gdzie są przechowywane klucze?

Aplikacja webowa trzyma odszyfrowane klucze w pamięci odblokowanej sesji. Blokada usuwa je z tej sesji. Zapisane lokalnie zaszyfrowane dane nadal wymagają klucza do odczytu.

Aplikacja mobilna nie jest jeszcze dostępna w sklepach. Jej obecny mechanizm szybkiego odblokowania może trwale zapisywać klucz główny pod ochroną systemu. Ten mechanizm wymaga przeglądu bezpieczeństwa i blokuje wydanie aplikacji; nie jest zatwierdzonym wyjątkiem od przechowywania kluczy tylko w pamięci.

Klucze tożsamości agenta są przechowywane w systemowym magazynie lub przez osobną usługę ochronną, zwaną brokerem. Różnicę między `Hardened` a `Convenience` opisuje instrukcja instalacji.

Tożsamość i odblokowanie

Logowanie a odblokowanie sejfu

Logowanie potwierdza, kim jesteś. Odblokowanie sejfu udostępnia aplikacji klucze do jego odczytania. To dwa osobne kroki: sama sesja logowania nie wystarcza do odszyfrowania danych.



Dwie niezależne przesłanki odczytu: uprawnienie do pobrania i zdolność odszyfrowania.

Odzyskiwanie

Klucz odzyskiwania pozwala odzyskać dostęp po utracie hasła głównego. Klient lokalnie otwiera chroniony materiał, pozwala ustawić nowe hasło i tworzy nowy klucz odzyskiwania. Zapisz nowy klucz w bezpiecznym miejscu.

Samo odzyskanie sesji logowania nie odszyfrowuje sejfu. Palladin nie może odtworzyć danych bez materiału potrzebnego do ich odszyfrowania.

Zmiana tożsamości

Zmiana hasła może zachować istniejący klucz prywatny i zabezpieczyć go nowym hasłem. Wymiana samej pary kluczy to inna operacja: wymaga ponownego przygotowania dostępu do udostępnionych sejfów.

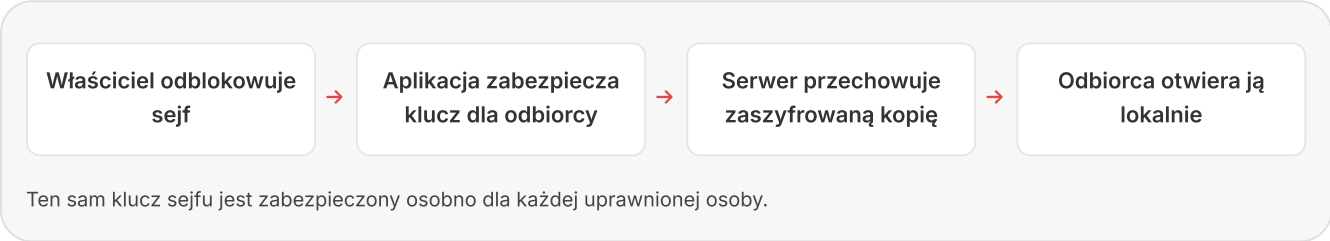
Granica klienta

Przeglądarka pobiera kod aplikacji z serwera. Jeśli ktoś podmieni ten kod, może próbować przechwycić dane podczas odblokowania. Dlatego szyfrowanie sejfu nie zastępuje bezpieczeństwa aplikacji i urządzenia.

Sejfy i udostępnianie

Jak udostępniasz sejf?

Udostępnienie sejfu daje odbiorcy uprawnienia oraz zaszyfrowaną kopię **klucza sejfu (VK, Vault Key)**. Odbiorca otwiera tę kopię swoim kluczem prywatnym. Nie otrzymuje Twojego klucza prywatnego ani hasła głównego.



Właściwy odbiorca

Aplikacja sprawdza tożsamość odbiorcy, jego klucz i członkostwo w sejfie. Samo zalogowanie do usługi nie wystarcza do podmienienia klucza lub uzyskania dostępu do danych.

Aktualizacja wpisu

Zmiana wpisu tworzy nową wersję danych. Kopie udostępniane agentom muszą odpowiadać tej wersji. Przygotowuje je uprawniona aplikacja użytkownika — serwer nie ma kluczy potrzebnych do odszyfrowania i przebudowania treści.

Usunięcie dostępu

Usunięcie członka blokuje kolejne pobrania. Aby chronić nowe dane przed osobą posiadającą stary klucz, trzeba zmienić klucz i zabezpieczyć dane dla pozostałych odbiorców. Nie usuwa to kopii już odczytanych przez usuniętego członka. Zobacz cofnięcie dostępu i rotację.

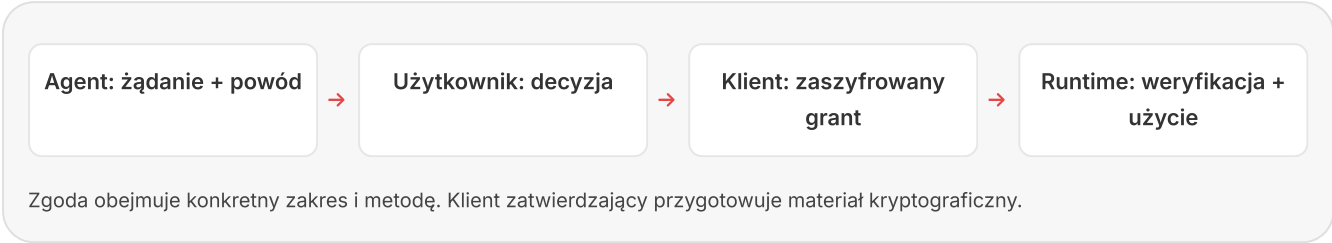
Tożsamość i granty Agenta

Tożsamość Agenta

Każdy agent ma własny identyfikator i klucze: jeden do odbierania zaszyfrowanych danych, drugi do podpisywania żądań. Profil to lokalna nazwa, pod którą wybierasz tego agenta. Klucz API łączy go z organizacją, ale nie zastępuje jego własnej tożsamości.

Parowanie przez przeglądarkę pozwala zatwierdzić połączenie bez kopiowania sekretu do modelu. Wewnętrzne poświadczenie przypisane do Agenta pozostaje pod logicznym kluczem organizacji i nie daje niezależnych uprawnień.

Zgoda na dostęp



Zgoda obejmuje konkretny zakres i metodę. Klient zatwierdzający przygotowuje materiał kryptograficzny.

Zgodę na dostęp nazywamy **grantem**. Określa ona zakres danych, sposób użycia, czas i limit operacji. Serwer sprawdza te warunki, a lokalny program Palladin sprawdza, czy otrzymane dane należą do właściwego agenta i zadania.

GRANULAR i FULL

Model	Materiał	Znaczenie
GRANULAR	Osobny klucz i kopia danych konkretnego wpisu	Agent otrzymuje tylko dane objęte zgodą, bez klucza całego sejfu.
FULL	Zaszyfrowany klucz sejfu (VK, Vault Key)	Agent otrzymuje klucz do sejfu. Wymaga to większego zaufania niż udostępnienie pojedynczego wpisu.

Limit pobrań kontroluje odpowiedzi serwera. Nie odbiera posiadaczowi klucza sejfu możliwości odszyfrowania danych, które już ma.

Metody i limity

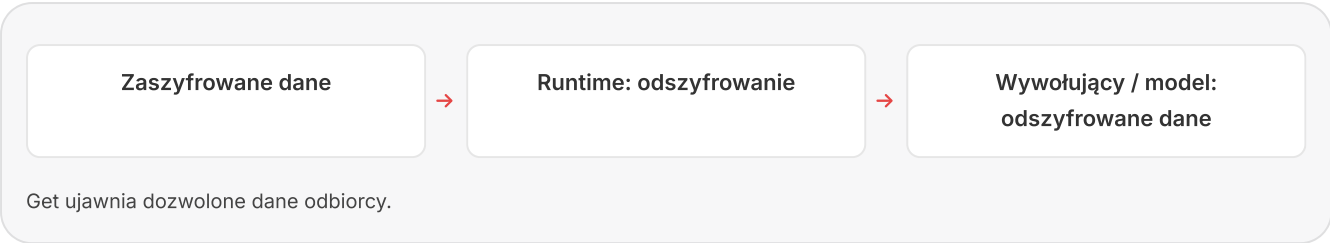
Grant określa dozwoloną metodę użycia. Uprawnienie `Exec` nie upoważnia do `Get`. Pobranie zaszyfrowanych danych może zużyć limit nawet wtedy, gdy uruchomiony program później zwróci błąd. Ponowienie może liczyć się jako kolejne użycie.

Odmowa i świadome cofnięcie dostępu są decyzją użytkownika. Agent nie powinien obchodzić ich przez nowy profil, inną metodę ani pętlę kolejnych próśb.

Użycie sekretu

Get: jawne ujawnienie

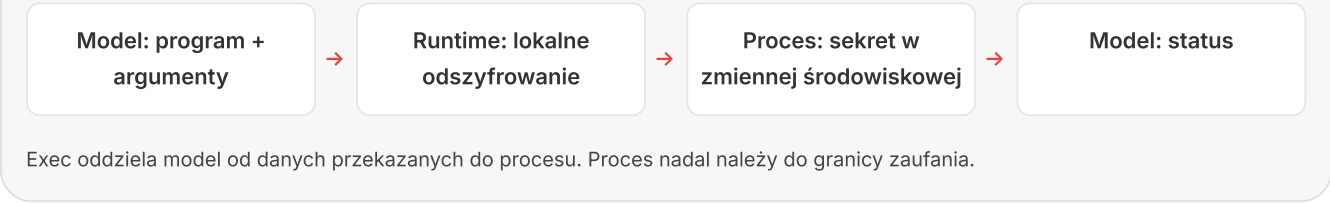
Runtime odszyfrowuje zatwierdzone dane i zwraca je wywołującemu. Jeśli wywołującym jest model, odszyfrowane dane trafiają do jego kontekstu. Nie należy opisywać tej ścieżki jako ukrywania sekretu przed AI.



Get ujawnia dozwolone dane odbiorcy.

Exec: użycie w procesie

Runtime przekazuje potrzebne pola do środowiska procesu docelowego. Nie umieszcza ich w argumentach polecenia. Standardowy wynik dla modelu jest ograniczony do statusu; wyjście procesu jest wstrzymywane. Proces docelowy otrzymuje jednak sekret i musi być zaufany w zakresie zadania.

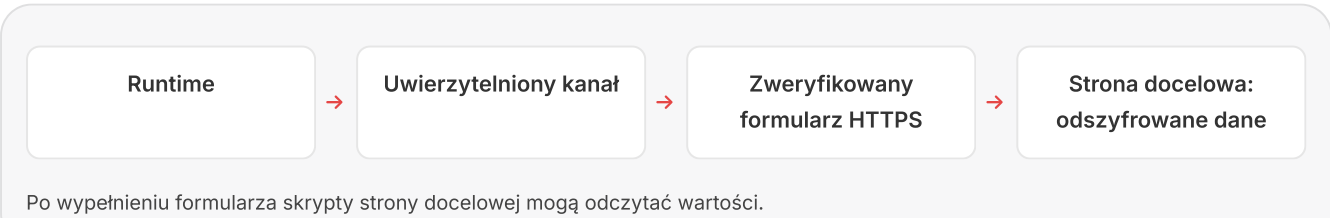


Exec oddziela model od danych przekazanych do procesu. Proces nadal należy do granicy zaufania.

Dla wpisów typu Script można osobno włączyć zwracanie wyniku skryptu do agenta. Taki wynik może ujawnić dane, również po ich zakodowaniu lub przekształceniu. Włączaj tę opcję tylko dla zaufanego skryptu i świadomie wybranego wyniku.

Inject: użycie w przeglądarce

Lokalny program Palladin przekazuje sekret do integracji przeglądarki przez zabezpieczone połączenie. Przed wypełnieniem formularza sprawdza adres HTTPS, kartę, aktualną stronę i wybrany wpis. Model otrzymuje wynik operacji, bez wartości pól.



Po wypełnieniu formularza skrypty strony docelowej mogą odczytać wartości.

Inject wymaga zgodnej integracji przeglądarki i uwierzytelnionego kanału do runtime. Sam identyfikator rozszerzenia nie dowodzi pochodzenia ze sklepu. Brak działającej integracji kończy operację. Nie powoduje ujawnienia hasła przez Get.

Czas życia danych

Runtime ogranicza czas i liczbę kopii odszyfrowanych danych. Nie zachowuje dostarczonych sekretów między wywołaniami. Nie jest to gwarancja bezpieczeństwa wobec uprzywilejowanego odczytu pamięci. Script na Unix ma dodatkowy, kontrolowany wyjątek: prywatny plik tymczasowy z kodem, usuwany po wykonaniu. Szczegóły zależą od platformy i artefaktu.

Cofnięcie dostępu i rotacja

Co daje cofnięcie dostępu?

Po cofnięciu lub wygaśnięciu zgody albo wykorzystaniu limitu serwer odmawia kolejnych pobrań. Przy każdym wydaniu danych sprawdza aktualne uprawnienia. Wcześniejszy status zapisany przez agenta nie wystarcza.

Czego nie można cofnąć?

Nie da się zmusić odbiorcy do zapomnienia hasła ani usunąć klucza, który wcześniej skopiował. Cofnięcie dostępu w Palladin nie zmienia też hasła w usłudze docelowej. Jeśli hasło lub klucz API wyciekły, zmień je u ich wystawcy.

Zmiana kluczy

Rotacja oznacza zastąpienie starego klucza nowym. Nowy **klucz sejfu (VK, Vault Key)** chroni przyszłe dane dopiero po odpowiednim ponownym zabezpieczeniu kluczy wpisów i danych oraz przekazaniu dostępu pozostałym odbiorcom. Materiał skopiowany przed zmianą pozostaje związany ze starymi kluczami.

Urządzenie bez internetu

Urządzenie offline nie otrzyma od razu informacji o cofnięciu dostępu. Serwer zablokuje kolejne pobrania, ale nie usunie danych ani kluczy już zapisanych na urządzeniu.

Zagrożenia i ograniczenia

Model zagrożeń

Zagrożenie	Mechanizm ochrony	Granica
Kradzież bazy zaszyfrowanych danych	Szyfrowanie klienckie i rozdzielenie kluczy	Metadane strukturalne nadal mogą być widoczne; siła hasła i sposób wyprowadzania klucza ma znaczenie.
Podmiana materiału między sejfami lub Agentami	Podpisy i niezależna sprawdzenie odbiorcy i kontekstu	Nie chroni przed wykonaniem złośliwego klienta.
Agent żąda nadmiernego dostępu	Decyzja użytkownika, GRANULAR, ograniczenie metody i czasu	Zatwierdzenie niebezpiecznego zakresu pozostaje decyzją o zaufaniu.
Prompt injection nakłania do użycia sekretu	Uprawnienia poza modelem i wybór minimalnego zakresu	Palladin nie rozpoznaje niezawodnie intencji modelu ani wszystkich szkodliwych zadań.

Zagrożenie	Mechanizm ochrony	Granica
Złośliwy proces docelowy Exec	Oddzielenie modelu i kontrolowana dostawa	Proces dostaje sekret i może go wykorzystać lub ujawnić.
Skrypt strony odczytuje wypełniony formularz	Ścisły wybór hosta i dokumentu przed zapisem	Strona docelowa obserwuje dane w swoim DOM.
Złośliwy proces tego samego użytkownika	Ochrona zależna od poziomu ochrony i systemu	Convenience nie izoluje innych procesów tego samego użytkownika.
Administrator/root odczytuje pamięć	Minimalizacja czasu życia i mechanizmy platformy	Przejęte urządzenie wykracza poza gwarancję szyfrowania na serwerze.
Cofnięty odbiorca zachował dane	Blokada nowych dostaw i rotacja	Nie można cofnąć wcześniejszej wiedzy.

FULL wymaga zaufania

FULL przekazuje klucz sejfu (VK, Vault Key) do lokalnego programu agenta. Pojedyncze odpowiedzi API, czas i licznik użyć są kontrolami online. Nie należy na ich podstawie obiecywać izolacji kryptograficznej od reszty sejfu wobec złośliwego posiadacza VK.

Dystrybucja jest częścią ochrony

Zmieniony launcher, runtime, web client lub provider może zmienić zachowanie systemu. Podpisy, pochodzenie artefaktów i zgodność wersji są częścią modelu, a nie dodatkiem do instalacji. Brak wymaganych dowodów powinien blokować wrażliwą operację.

Co oznacza „bez sekretu w modelu”

Dotyczy konkretnej ścieżki dostawy. Nie obejmuje Get, jawnie zwracanych wyników Script ani danych odczytanych później z procesu lub strony. Ukrycie wartości przed modelem nie usuwa ryzyka wykonania niepożądanej operacji przy użyciu tej wartości.

Weryfikowalność i wydania

Źródła techniczne

Źródło	Zakres
palladin-agent	CLI, MCP, natywny runtime i kontrakty platform.
palladin-crypto	Algorytmy i kod szyfrowania współdzielony przez aplikacje.
palladin-protocol	Opis protokołu i wspólne dane testowe.

Źródło	Zakres
palladin-security	Dokumentacja zapewnienia bezpieczeństwa.

Jak czytać ten dokument

Whitepaper opisuje architekturę, przepływ kluczy i granice ochrony Palladin. Kod protokołu i testy zgodności uzupełniają ten opis. Raport z audytu jest osobnym dokumentem: jego wnioski dotyczą wskazanej wersji i zakresu badania.

Historia dokumentu

Wersja	Data	Zmiana
1.0	2026-10-25	Model bezpieczeństwa, klucze, dostęp agentów i ograniczenia ochrony.

Wersje PL i EN oraz formaty HTML, Markdown i PDF opisują ten sam model. Numer whitepapera jest numerem dokumentu, niezależnym od wersji CLI.

Aneks: kryptografia

Ten aneks zawiera szczegóły techniczne. Wyjaśnienie kluczy znajdziesz w rozdziale Dane i klucze. VK oznacza klucz sejfu (Vault Key), a DEK — klucz szyfrowania danych (Data Encryption Key).

Zestaw kryptograficzny

Operacja	Mechanizm
Wyprowadzenie z hasła	Argon2id: 32 MiB pamięci, 2 iteracje, 1 wątek, 32-bajtowy sekret bazowy konta; profil `identity-argon2id-password-v1`.
Oddzielenie uwierzytelnienia od klucza głównego	HKDF-SHA256 wyprowadza z sekretu bazowego dwa klucze, używając różnych etykiet przeznaczenia i identyfikatora konta.
Ochrona klucza prywatnego konta	XSalsa20-Poly1305. Odzyskiwanie konta używa osobnego profilu Argon2id z 19 MiB pamięci.
Szyfrowanie treści sejfu i zabezpieczanie kluczy symetrycznych	XChaCha20-Poly1305, suite `palladin-vault-xchacha-v1`.
Asymetryczne przekazanie klucza	X25519 sealed package, suite `palladin-x25519-sealed-box-v1`.
Podpisy tożsamości Agenta	Ed25519.
Klucze symetryczne VK / DEK	Losowe 32 bajty, generowane po stronie klienta.

Nazwy algorytmów nie wystarczają do odtworzenia protokołu. Implementacja musi stosować dokładny format, domeny podpisów, kodowanie, reguły nonce i kanonikalizację z wersjonowanego kontraktu. Nie implementuj kryptografii na podstawie samego tego streszczenia.

Walidacja przed użyciem

1. Odczytaj jawny identyfikator suite i wersję protokołu; odrzuć nieobsługiwany wariant.
2. Sprawdź format i ograniczenia rozmiaru przed przetwarzaniem.
3. Zweryfikuj podpis oraz wymagane commitmenty.
4. Porównaj organizację, sejf, odbiorcę, przyznany dostęp, rewizje, epoki i wersje kluczy z niezależnym, zaufanym źródłem.
5. Otwórz materiał tylko w dozwolonej granicy i przerwij przy błędzie uwierzytelnienia.

Zgodność implementacji

Wspólne dane i wektory testowe powinny sprawdzać zgodność klientów i backendu. Testy negatywne muszą obejmować podmienionego odbiorcę, inny sejf, starą rewizję, niewłaściwą epokę i zmodyfikowane zaszyfrowane dane. Nie wystarcza test, w którym biblioteka odszyfrowuje własny wynik.