

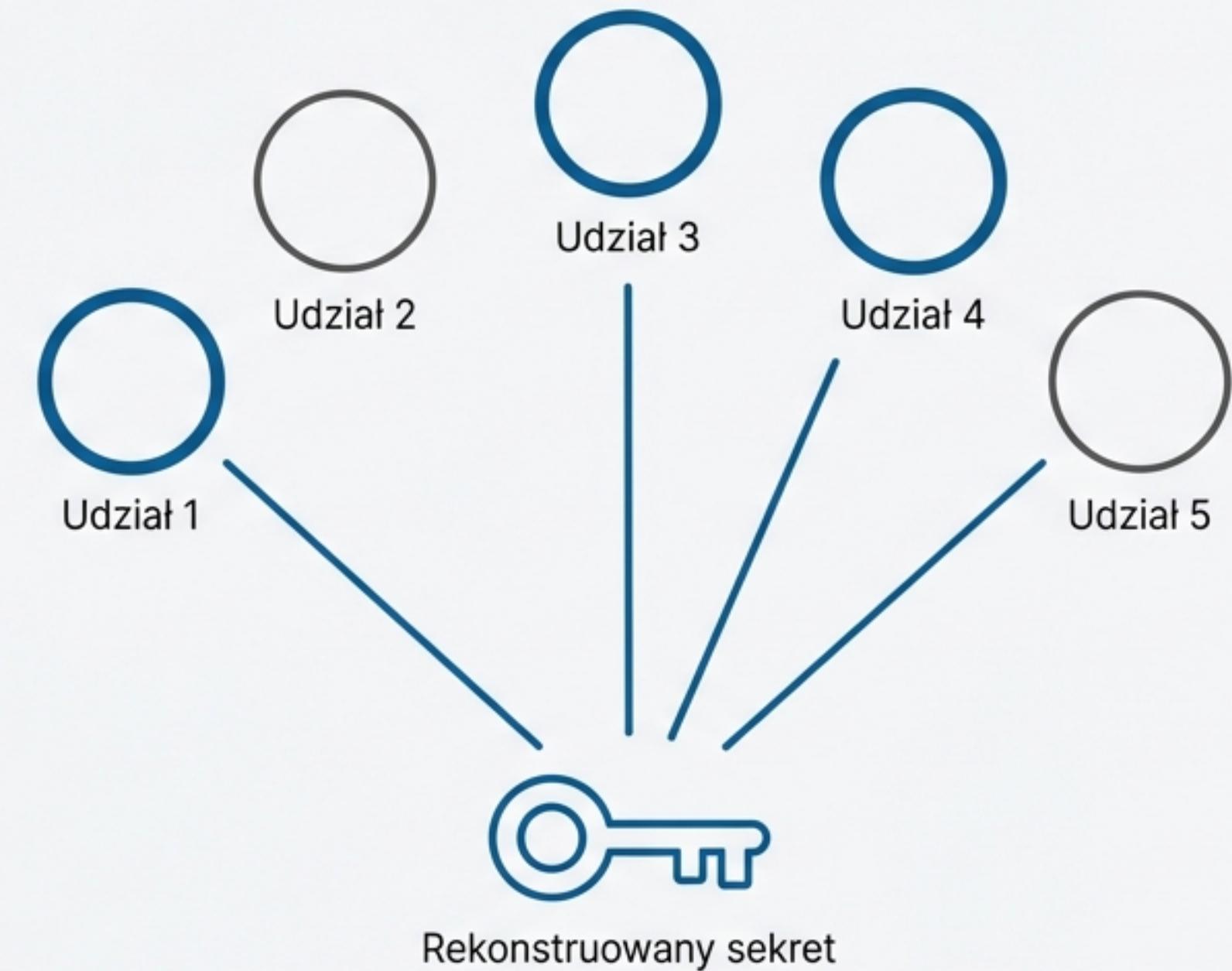
Problem: Pojedynczy Punkt Awarii

- Wyzwanie: Jeden klucz, jedna osoba, jeden serwer. Co, jeśli ten pojedynczy element zawiedzie?
- Kompromis bezpieczeństwa i niezawodności: Przechowywanie jednej kopii jest ryzykowne (awaria, utrata). Wiele kopii zwiększa ryzyko kradzieży.
- Potrzeba: Metody podziału danych D na n części, gdzie dowolne k części pozwala na odtworzenie całości, ale $k-1$ części nie ujawnia absolutnie nic.
- Pionierskie rozwiązanie: Schemat Dzielenia Sekretu autorstwa Adi Shamira (1979).



Rozwiązanie: Schemat Progowy (k, n)

- **N**: Całkowita liczba wygenerowanych i rozdysytrybuowanych udziałów (części sekretu).
- **K**: Minimalna liczba udziałów (próg) potrzebna do odtworzenia oryginalnego sekretu.
- **Zasada #1**: Dowolne 'K' lub więcej udziałów pozwala w pełni zrekonstruować sekret.
- **Zasada #2**: Dowolne 'K-1' lub mniej udziałów nie ujawnia **żadnej** informacji o sekrecie. Wartość sekretu pozostaje „całkowicie nieokreślona”.
- **Przykład**: Schemat `(3, 5)` oznacza, że sekret jest dzielony na 5 udziałów, a do jego odtworzenia potrzebne są dowolne 3 z nich.



Gwarancja: Doskonała Tajność

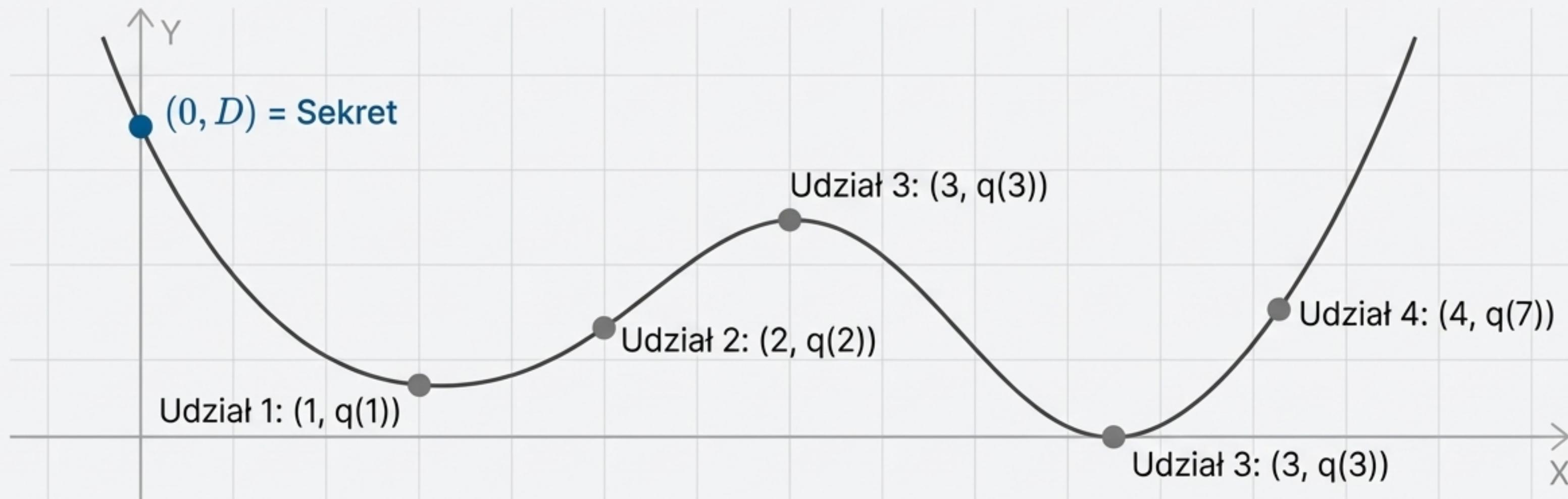
- Posiadanie ' $K-1$ ' udziałów nie daje absolutnie żadnej przewagi w odgadnięciu sekretu. Wszystkie możliwe wartości pozostają jednakowo prawdopodobne.
- To jest **bezpieczeństwo informacyjno-teoretyczne**, w przeciwieństwie do bezpieczeństwa obliczeniowego (np. RSA, AES).
- Oznacza to, że nawet nieskończona moc obliczeniowa nie jest w stanie złamać sekretu, mając mniej niż ' K ' udziałów.
- Gwarancja ta wynika bezpośrednio z matematycznych właściwości wielomianów.



Bezpieczeństwo Informacyjno-Teoretyczne
vs. Bezpieczeństwo Obliczeniowe

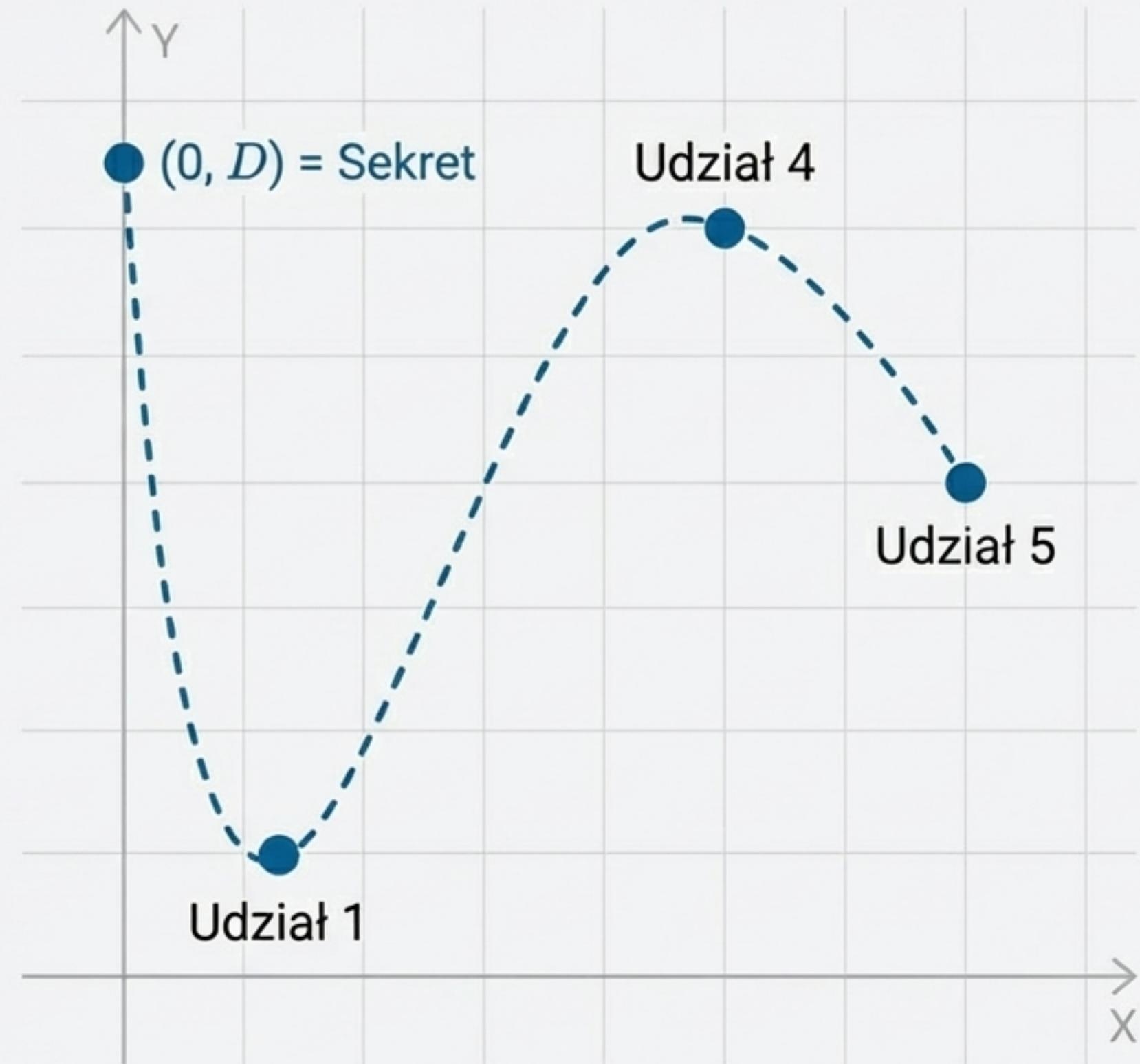
Krok 1: Jak Dzielimy Sekret za Pomocą Wielomianu

- **Sekret staje się punktem na osi:** Traktujemy nasz sekret D jako liczbę. Staje się on wyrazem wolnym a_0 w wielomianie, czyli punktem $(0, D)$.
- **Generujemy losową krzywą:** Tworzymy losowy wielomian $q(x)$ stopnia $k - 1$:
$$q(x) = a_0 + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1}$$
Współczynniki $a_1 \dots a_{k-1}$ są losowe, co zapewnia losowość krzywej.
- **Rozdajemy punkty z krzywej:** Każdy udział to punkt na tym wielomianie. Uczestnik i otrzymuje parę wartości $(i, q(i))$.



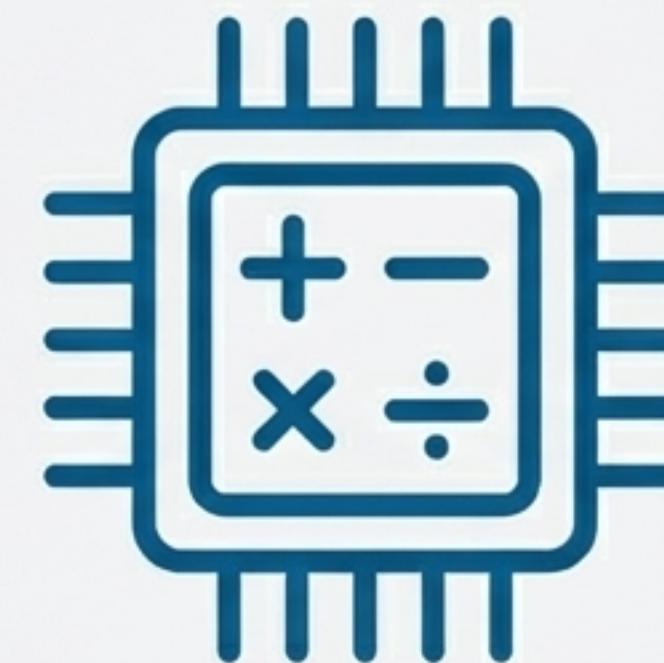
Krok 2: Jak Odtwarzamy Sekret przez Interpolację

- 1. **Zbieramy punkty***: Potrzebujemy co najmniej K udziałów (czyli K punktów (x, y)).
- 2. **Odtwarzamy unikalną krzywą***: Zgodnie z podstawowym twierdzeniem algebry, K punktów jednoznacznie definiuje wielomian stopnia $k - 1$. Proces ten nazywa się interpolacją.
- 3. ***Znajdujemy sekret na osi***: Po zrekonstruowaniu wielomianu $q(x)$, obliczamy jego wartość dla $x = 0$. Wynik $q(0)$ to nasz oryginalny sekret D .



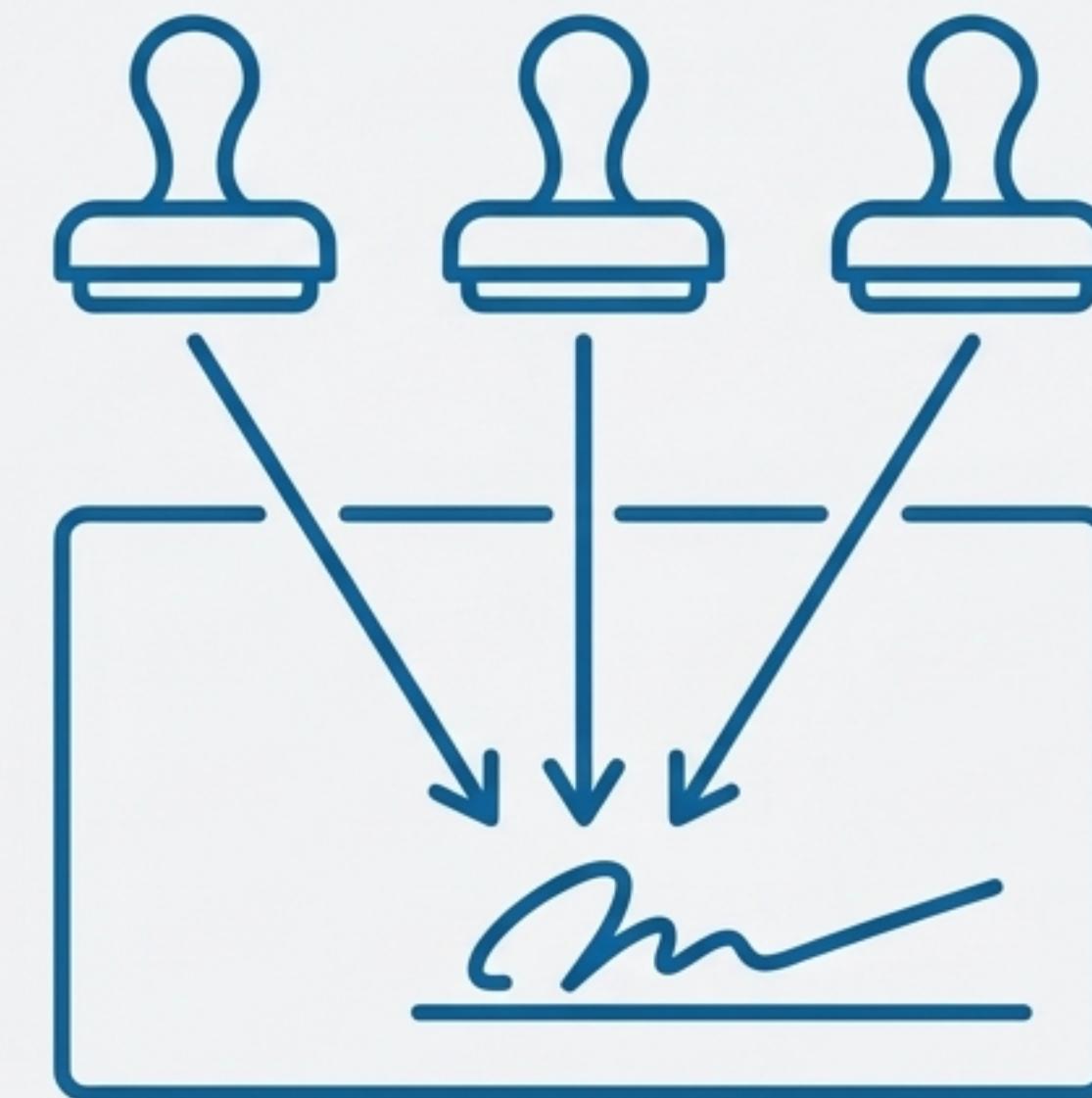
Wymagania Praktyczne i Wydajność

- **Precyza:** Obliczenia wymagają arytmetyki dokładnej. Rozwiązaniem jest arytmetyka modularna.
- **Ciąła Skończone:** Wszystkie operacje są wykonywane modulo p , gdzie p jest liczbą pierwszą większą niż jakikolwiek sekret (D) i liczba udziałów (n).
- **Wydajność:** Algorytmy interpolacji i ewaluacji wielomianów są bardzo szybkie.
- **Brak Kosztownych Operacji:** W przeciwieństwie do RSA, schemat nie wymaga kosztownych obliczeniowo operacji potęgowania.



Klasyczny Przykład: Podpisywanie Czeków w Firmie

- **Scenariusz (z pracy Shamira, 1979):** Firma chce cyfrowo podpisywać wszystkie swoje czekи, używając tajnego klucza 'D'.
- **Problem:** Dawanie klucza każdemu menedżerowi jest ryzykowne. Wymaganie zgody wszystkich jest niepraktyczne.
- **Rozwiązanie:** Schemat progowy $(3, n)$, gdzie 'n' to liczba menedżerów.
- **Implementacja:** Każdy menedżer otrzymuje jeden udział D_i . Urządzenie podpisujące akceptuje dowolne 3 udziały, aby tymczasowo odtworzyć klucz 'D' i złożyć podpis.
- **Wynik:** Żaden pojedynczy menedżer nie może samodzielnie fałszować podpisów. Potrzebna jest współpraca co najmniej dwóch wspólników.



Wzmacnianie Bezpieczeństwa: Proaktywne Odświeżanie Udziałów

- W oryginalnej pracy Shamir zauważył, że można „łatwo zmienić części D_i bez zmiany oryginalnych danych D ”.
- **Mechanizm:** Wystarczy wygenerować nowy, losowy wielomian $q'(x)$ z tym samym wyrazem wolnym ($a_0 = D$).
- **Proces:** Okresowo generuje się i rozdaje nowe udziały z nowej krzywej.
- **Efekt:** Stare udziały stają się bezużyteczne. Atakujący, który zbierał udziały przez długi czas, musi zacząć od nowa. Bezpieczeństwo jest zachowane.



Zastosowania Współczesne: Gdzie Dziś Działa Schemat Shamira



Portfele Kryptowalut: Mechanizmy odzyskiwania dostępu do portfeli sprzętowych i cyfrowych (np. „Shamir Backup”).



Zarządzanie Kluczami w Chmurze: Dzielenie głównego klucza szyfrującego pomiędzy różnych, niezależnych dostawców chmury.



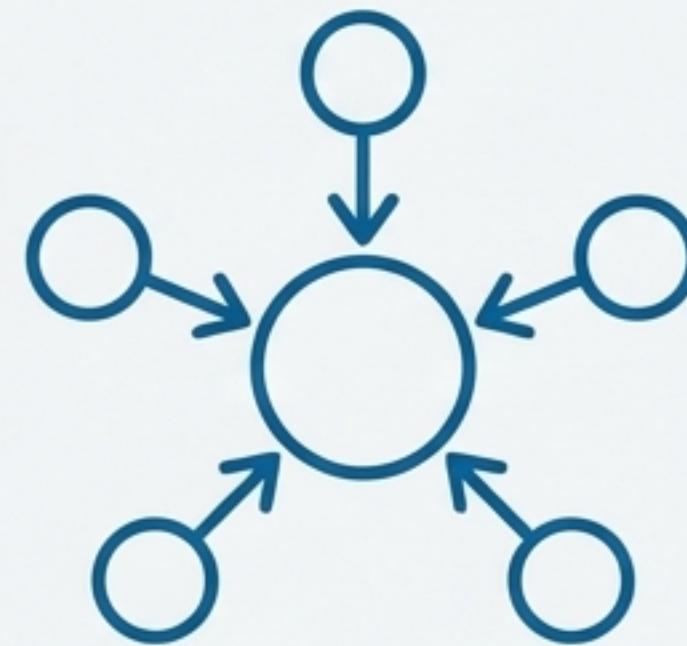
Menedżery Haseł: Bezpieczne mechanizmy odzyskiwania konta dla zespołów i rodzin, eliminujące pojedynczy „master password” jako punkt awarii.



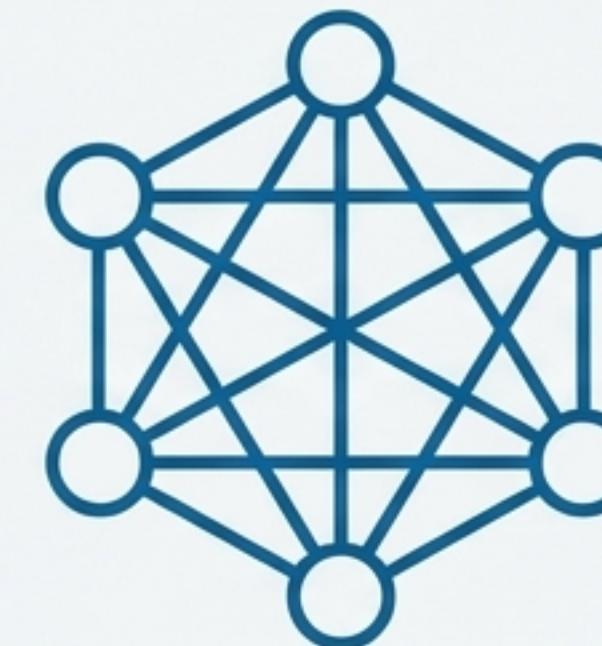
Systemy Zdecentralizowane (Blockchain): Zarządzanie uprawnieniami w organizacjach typu DAO i wykonywanie operacji na smart kontraktach.

Zmiana Paradygmatu: Od Centralizacji do Rozproszonego Zaufania

Stary Model: Centralizacja



Nowy Model: Dystrybucja



- Zaufaj jednej osobie, jednemu serwerowi, jednej firmie.

- Rozprosz zaufanie pomiędzy wielu uczestników, gdzie żaden z nich nie ma pełnej kontroli.
- Praktyczna realizacja zasady „nie ufaj, weryfikuj” na poziomie architektury systemu.
- Zamiast eliminować ryzyko, inteligentnie nim zarządzamy, równoważąc bezpieczeństwo, niezawodność i dostępność.

Pytanie do Ciebie

Czy krytyczne systemy w bankach, rządach i korporacjach powinny wdrożyć rozproszone dzielenie sekretów, zamiast ufać pojedynczym administratorom?