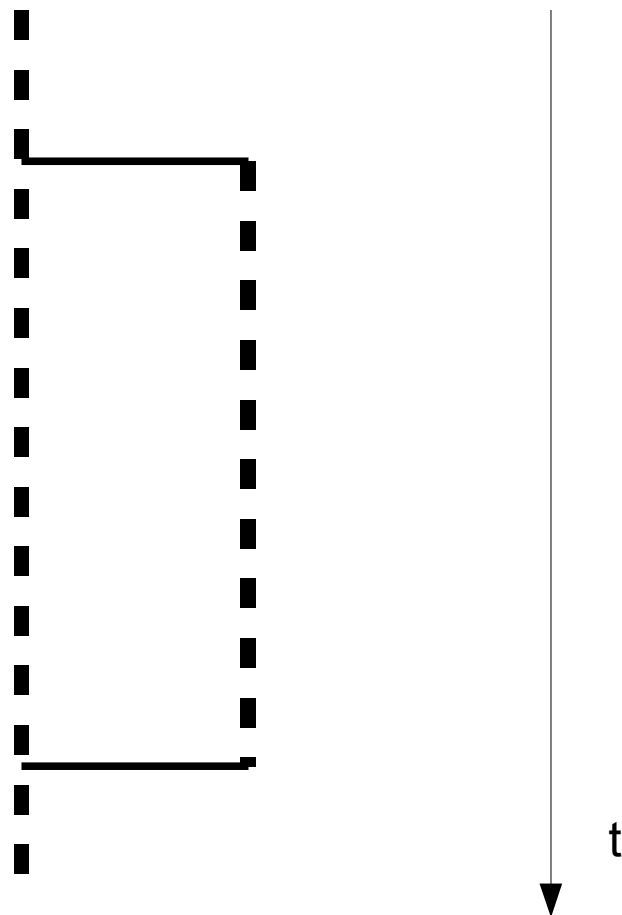
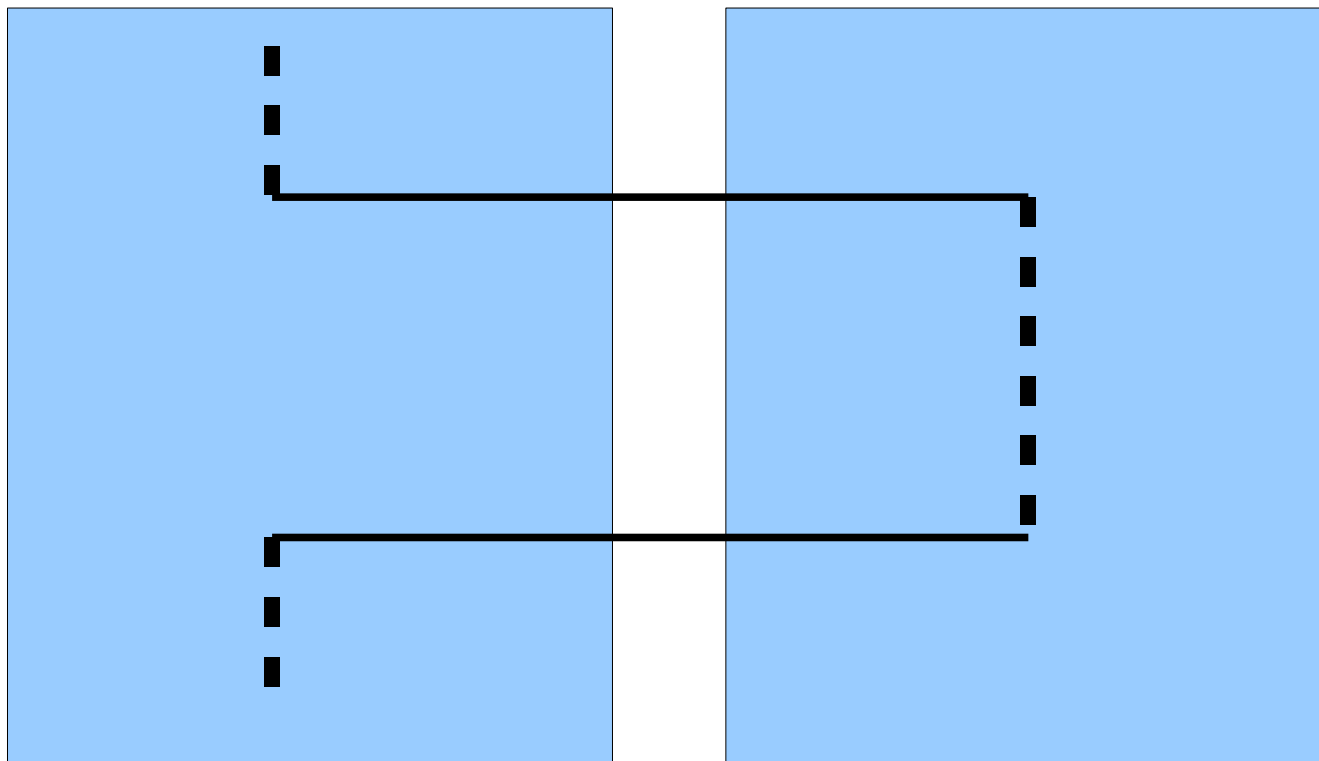


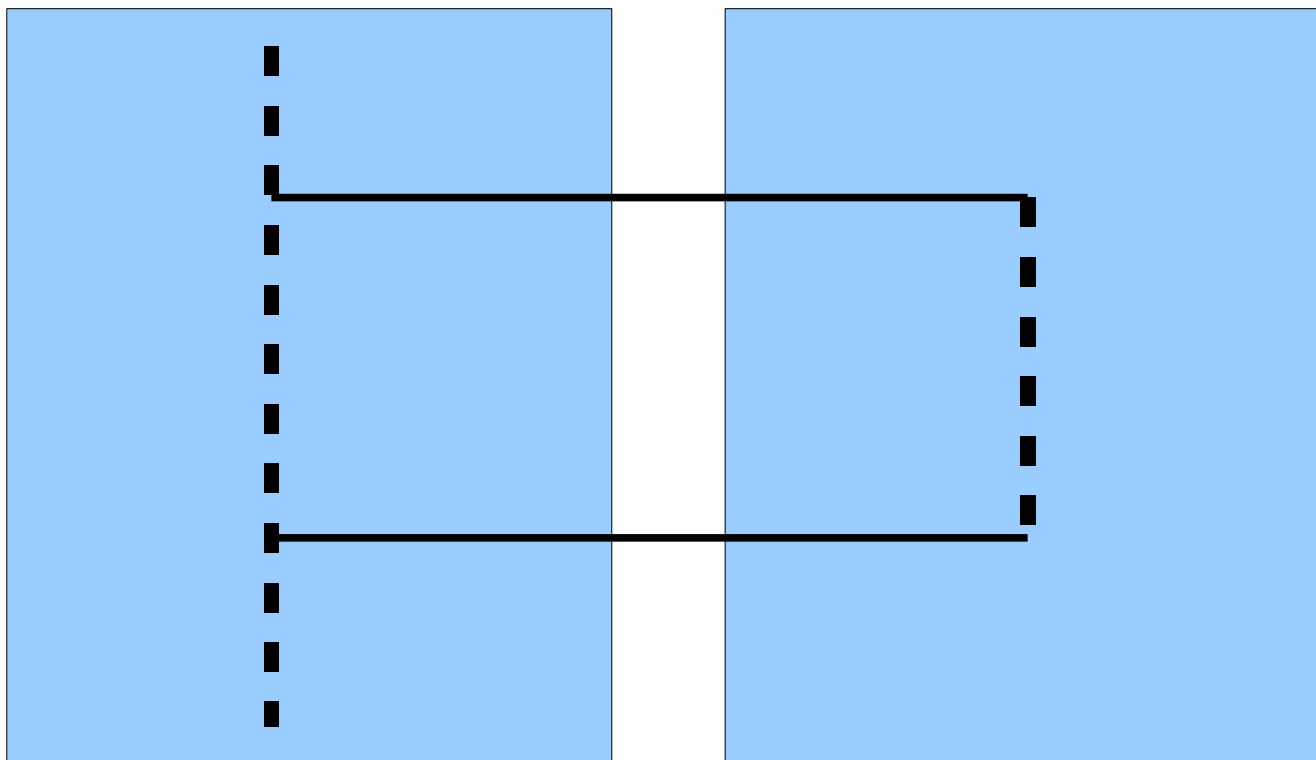
Program współbieżny



Program rozproszony



Program rozproszony i współbieżny



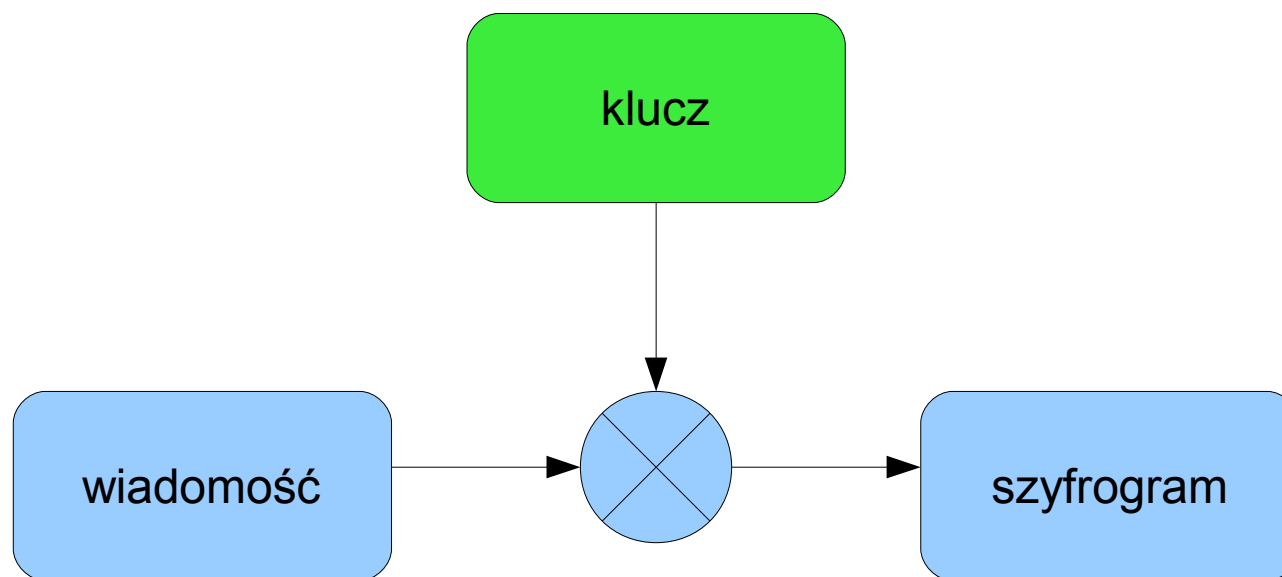
Rozpraszanie

- Poprawa wydajności.
Pozwala uzyskać współbieżność na komputerach sekwencyjnych. Zbiór komputerów połączonych siecią tworzy wirtualny komputer równoległy w architekturze typu message passing.
- Dostępność zasobów.
Część programu działa w lokalizacji, w której dostępne są zasoby niezbędne do wykonania zadania.

Założenia

- Przezroczystość
Wywołania zdalne powinny w jak największym stopniu przypominać wywołania lokalne.
- Wydajność
Narzut samego wywołania zdalnego podprogramu powinien być jak najmniejszy.
- Reprezentacja
Dane powinny być poprawnie interpretowane po obu stronach wywołania.

Przykład 1 – łamanie szyfru



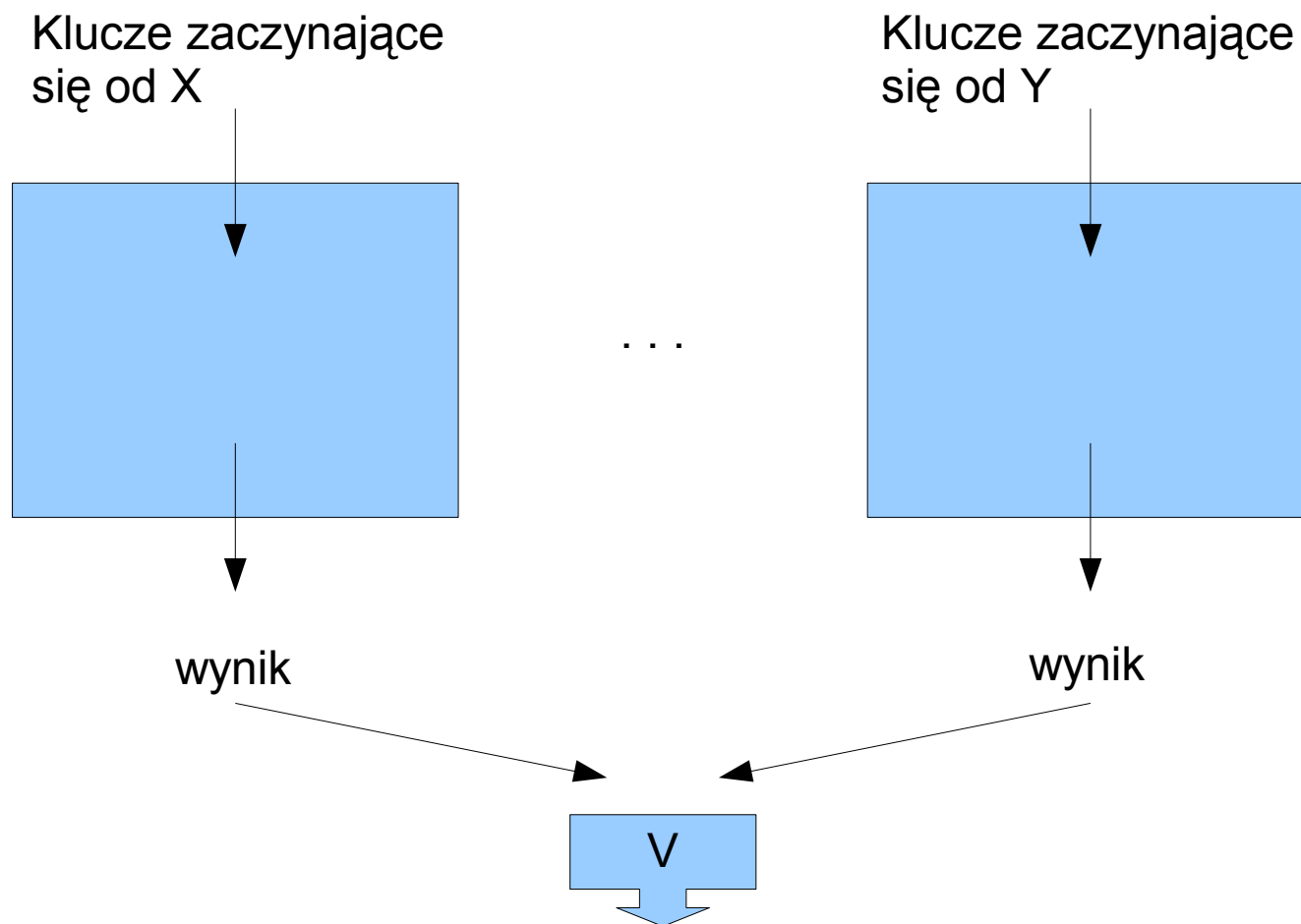
Łamanie szyfru

- Dane:
jawny tekst
szyfrogram
[długość klucza]
- Szukane:
klucz, którym zakodowano wiadomość do postaci szyfrogramu

Metoda brute force

- klucz := pierwszy klucz
- while klucz <> null do
 - if odszyfruj(szyfrogram, klucz) == wiadomość
then return klucz
else klucz := następny klucz
- return „nie znaleziono”

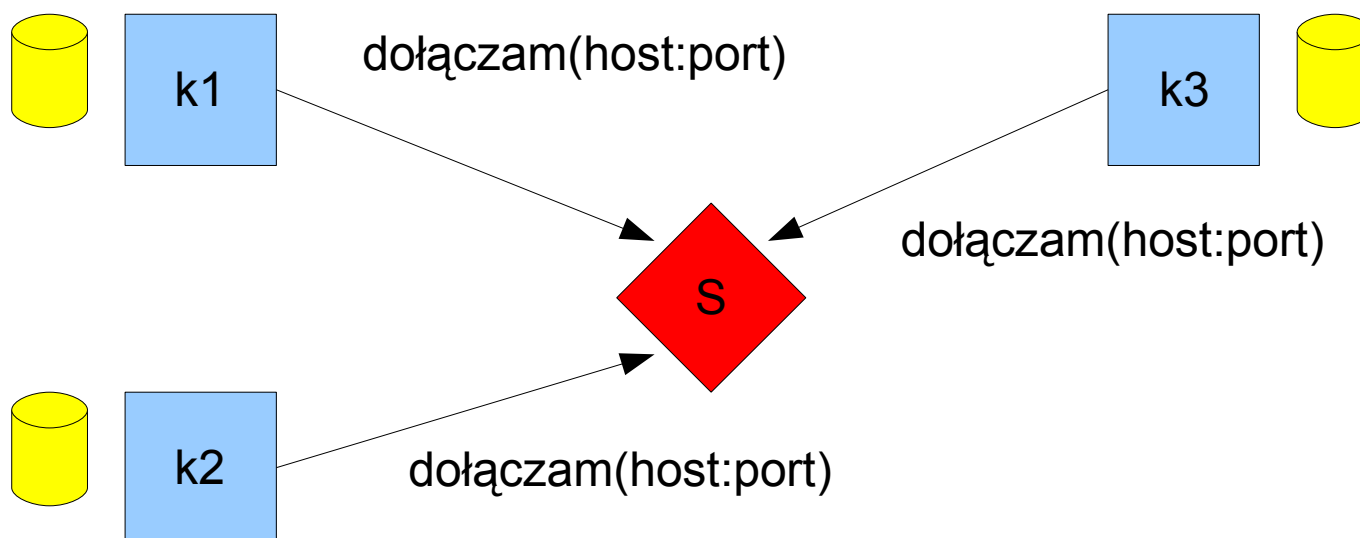
Zrównoleglenie brute force



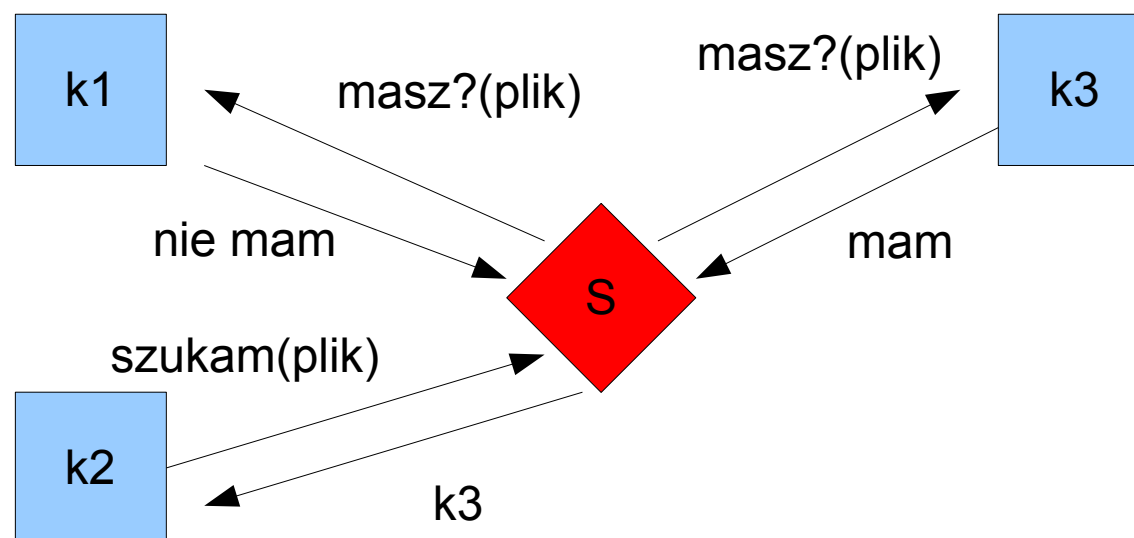
Zrównoleglenie i rozproszenie

- `pocz_klucz := pierwszy początek klucza`
- `repeat`
 - sprawdź czy któryś z serwerów znalazł klucz
 - `serwer := pierwszy wolny serwer`
 - zleć szukanie klucza o początku `pocz_klucz` serwerowi `serwer`
 - `pocz_klucz := następny początek klucza`
- `until znaleziono klucz or pocz_klucz == null`

Przykład 2 - P2P



P2P – szukam pliku



P2P – pobieram plik

