



# Systemy Operacyjne

**Bezpieczeństwo**

Dr inż. Krzysztof Rzecki

Na podstawie: Abraham Silberschatz, *Koncepcje systemów operacyjnych*



# Bezpieczeństwo

- **Bezpieczeństwo** jest miarą ufności, że integralność systemu i danych zostanie zachowana.
- Cztery najważniejsze funkcje bezpieczeństwa:
  - Uwierzytelnianie / Autentyczność
  - Poufność
  - Integralność
  - Rozliczalność / Niezaprzeczalność.
- Zasoby systemu komputerowego:
  - Informacje (dane, kod)
  - Czas procesora
  - Pamięć główna, pamięć zapasowa
  - Dostęp do sieci komputerowej
  - etc.



# Aspekty biznesowe

- Utrata informacji przez jej bezpowrotne usunięcie.
- Kompromitacja informacji przez jej ujawnienie.
- Wykorzystanie zasobów przedsiębiorstwa, np do rozsyłania spamu, tzw. kopania kryptowalut lub ataku na inne systemy komputerowe.



# Problem bezpieczeństwa

- System jest **bezpieczny**, kiedy dostęp do jego zasobów oraz ich wykorzystanie odbywa się zgodnie z ustalonym przeznaczeniem.
- Z jednej strony można wprowadzać coraz mocniejsze zabezpieczenia.
- Z drugiej strony, zabezpieczenia nie mogą powodować uciążliwości w korzystaniu z systemu (patrz: odłączenie od sieci internet).
- Dobór zabezpieczeń (i ich koszt) musi być dopasowany do wartości przechowywanych informacji, czy krytyczności działania systemu oraz potencjalnych zagrożeń (nie ma sensu wprowadzać firewall do systemu bez dostępu do internetu)



# Zarządzanie informacją

- Pozyskiwanie
- Parsowanie
- Oczyszczanie
- Transformacja
- Przetwarzanie
- Przechowywanie
- Backup
- Archiwizacja
- Przesyłanie
- Dodawanie
- Zmiana
- Usuwanie



# Pozyskiwanie

- Ang. *acquiring* – pobieranie do systemu danych ze źródeł zewnętrznych.
- Źródła pozyskiwania danych:
  - Skaner dokumentów papierowych
  - Sieć komputerowa (inne hosty)
  - Urządzenie pomiarowe cyfrowe
  - Urządzenia akwizycji sygnałów analogowych
  - itp.



# Parsowanie

- Ang. *parsing* – proces wyuskiwania interesujących nas danych z ciągów znaków, obrazów, sygnałów, itp.
- Parsowanie tekstu realizowane jest najczęściej w technice wyrażeń regularnych (ang. *regular expression*).
- Przykład: Dziś jest piątek trzynastego. Jeśli parsowanie ma wyuskać dni tygodnia to z tego ciągu otrzymamy 'piątek'.



# Oczyszczanie

- Ang. *cleaning* – proces markowania danych, które są błędne.
- Oczyszczanie realizowane jest podobnie do parsowania.
- Przykład: Skanning laserowy prowadzony z samolotu-awionetki w rozdzielczości 12 punktów na 1 m<sup>2</sup> wykazał na obszarze oznaczonym jako jezdnia pewne osoby. Jeśli celem jest stworzenie ortofotomapy, to dane związane z rozpoznanymi osobami oznaczyć należy jako błędne.





# Transformacja

- Ang. *transforming* – proces przekształcania danych między formatami.
- ETL – ang. Extract Transform Load – ciąg trzech procesów wyciągania danych z bazy danych źródłowych, transformacji do innego formatu i załadowanie do bazy danych docelowych.
- Przykład: Pobranie danych ze strony WWW MPK, przekształcenie na strefę czasową UTC i załadowanie do bazy MySQL.



# Przetwarzanie

- Ang. *processing* – obróbka danych za pomocą algorytmów charakterystycznych dla danego problemu.



# Przechowywanie

- Ang. *storing* – przechowywanie danych na nośnikach o dostępie bezpośrednim (ang. direct access).
- Przechowywanie w warstwie:
  - W warstwie fizycznej (dane binarne)
  - W warstwie logicznej systemu plików
  - W warstwie logicznej bazy danych



# Backup i archiwizacja

- Ang. *backeping* – sposób utworzenia kopii danych w celu szybkiego ich odtworzenia na wypadek wystąpienia awarii.
- Ang. *archiving* – sposób utworzenia kopii danych w celu długotrwałego przechowywania.
- Archiwizacja:
  - Pełna (ang. *mirroring*),
  - Przyrostowa (ang. *incremental*).



# Przesyłanie

- Ang. *transferring* – tworzenie kopii danych w innym systemie komputerowym za pomocą sieci komputerowej.
- Szybkość przesyłania liczona jest w bitach (kilobitach, megabitach, gigabitach) na sekundę.
- Przykłady:
  - $1 \text{ b/s} = 1 \text{ bps}$
  - $1024 \text{ b/s} = 1 \text{ Kbps}$
  - $1048576 \text{ b/s} = 1 \text{ Mbps}$



# Zarządzanie bezpieczeństwem

- Poufność
- Integralność
- Dostępność
- Rozliczalność
- Identyfikacja / Uwierzytelnianie
- Autoryzacja
- Awaria / Niezawodność
- Anonimowość
- Zagrożenie
- Ryzyko
- Podatność
- Zabezpieczenie
- Monitorowanie
- Odtwarzanie



# Poufność

- Ang. *confidentiality* –ochrona danych przed ich ujawnieniem osobom i/lub procesom.
- Poufność realizowana jest przez szyfrowanie danych i/lub kanałów komunikacji.
- Szyfrowanie: symetryczne i asymetryczne.



# Integralność

- Ang. integrity - rozpoznanie zmiany, dodania lub usunięcia danych.
- Narzędziem do badania integralności są sumy kontrolne, kody korekcyjne CRC, kody MAC.
- Zaawansowane badanie integralności opiera się o podpis cyfrowy.





# Dostępność

- Ang. *availability* – własność danej polegająca na tym, że jest ona dostępna dla osoby / procesu w zadanym przedziale czasu i zadanym miejscu.
- Dostępność to także stosunek czasu bezawaryjnego działania danej usługi w odniesieniu do całości założonego czasu.



# Rozliczalność / Niezaprzeczalność

- Ang. *accountability* / ang. *nonrepudiation* – zapewnienie, że aktywność danej osoby / procesu może zostać bezsprzecznie stwierdzona.
- Rozliczalność realizowana jest za pomocą logowania (rejestrowanie zdarzeń).



# Identyfikacja / Uwierzytelnianie

- Ang. *authenticity* – weryfikacja osoby / procesu.
- Weryfikację poprzedza identyfikacja (ang. *identification*).
- Weryfikacja realizowana jest za pomocą loginu i hasła, haseł jednorazowych, albo technik biometrycznych.

Zobacz: <https://doi.org/10.1016/j.ins.2017.05.041>



# Autoryzacja

- Ang. *authorization* - przyznanie uprawnień.



# Awaria

- Ang. *malfunction* – stan niesprawności systemu uniemożliwiający jego normalne użytkowanie i działanie.
- Awaria definiowana jest zwykle jako nagła i nieprzewidywalna (choć mogą wystąpić oznaki wskazujące zbliżającą się awarię).
- Awaria dotyczyć może całego systemu, lub niektórych funkcjonalności (zwykle kluczowych).



# Niezawodność

- Ang. reliability – własność obiektu / systemu stwierdzająca prawdopodobieństwo nie wystąpienia awarii:
  - $R(t) = P\{t \geq r\}$
  - $R(t)$  – niezawodność po czasie  $t$
  - $r$  – założony czas pracy
- $\lim_{t \rightarrow \infty} R(t) = 0$
- Niezawodność  $R(1h) = 90\%$  oznacza, że w pierwszej godzinie wystąpi 10% awarii.
- Niezawodność realizowana jest przez redundancję i nadmiarowość.



# Anonimowość

- Ang. *anonymity* – własność związana z brakiem możliwości identyfikacji osoby / procesu, czy też powiązania zdarzenia, utworu z osobą / procesem.
  - Anonimowość w sieci realizowana jest przez serwery proxy, NAT, sieć ‘cebulową’.
  - Anonimowość technicznie nie jest możliwa – ale jest możliwa prawnie.
- 
- Anonimizacja danych - jednokierunkowa zmiana wartości identyfikacyjnych.



# Zagrożenie i ryzyko

- Ang. *threat* – stan obniżonego bezpieczeństwa.
  - Ang. *risk* – prawdopodobieństwo, że zagrożenie zmieni stan osoby / procesu / zasobu / systemu z pozytywnego w negatywny (w tym w awarię).
- 
- Zagrożeniem jest włamanie, a z uwagi na hasła słownikowe ryzyko jest wysokie.
  - Zagrożeniem jest uszkodzenie dysku (awaria), ale redundancja obniża ryzyko.





# Podatność i zabezpieczenie

- Ang. *susceptibility* – wysokie ryzyko zmiany zagrożenia w awarię.
- Ang. *protection* – obniżanie ryzyka zmaterializowania zagrożenia (awarii).
  
- Podatny system na włamania można zabezpieczyć uruchamiając firewall.



# Monitorowanie

- Ang. *monitoring* – realizacja procesów obserwacji osoby / procesu / systemu o charakterze ciągłym i długotrwałym.
- Monitoring realizowany jest przez ciągłe analizowanie logów (rozliczalność), odpytywanie o stan, obserwację bezinwazyjną.



# Odtwarzanie

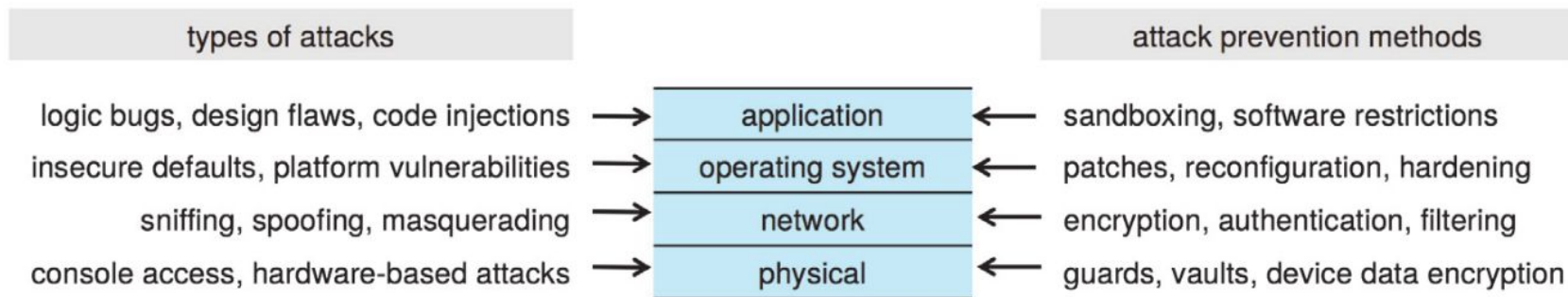
- Przywracanie stanu systemu sprzed awarii.
- Przywracanie może być pełne lub częściowe.
- Przywracanie może wyłączać system z użycia lub trwać w trakcie jego pracy.



# Podstawowe typy ataków

- Naruszenie poufności, integralności, czy dostępności do danych.
- Kradzież usługi, np. przez przechwycenie danych uwierzytelniających.
- Odmowa usługi (ang. *Denial-of-service*, DOS) oraz DDOS (ang. *distributed denial of service*).
- Powtarzanie operacji (ang. *reply attack*), czyli wykonanie powtórne tej samej operacji (np. przelew).
- Atak ang. *man-in-the-middle* polegający na umieszczeniu atakującego pomiędzy stronami.
- Atak ang. *session hijacking* polegający na przechwyceniu sesji (może poprzedzać powyższy atak).
- Eskalacja uprawnień (ang. *privilege escalation*) - przekazywanie i rozszerzanie uprawnień.

# Cztery poziomy bezpieczeństwa



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*

Jednak najsłabszym ogniwem zwykle jest: **człowiek** (socjotechnika, ang. *social engineering*).



# Oprogramowanie naruszające bezpieczeństwo

- **Malware** - oprogramowanie przeznaczone do wykorzystania, zablokowania lub uszkodzenia systemu komputerowego. Podstawa działania: uruchamianie z uprawnieniami innego użytkownika.
- **Koń trojański** - oprogramowanie realizujące w ukryty sposób szkodliwą funkcjonalność, na przykład: pobieranie informacji o danych do logowania, kontakty użytkownika, itp.
- **Spyware** - odmiana konia trojańskiego, którego funkcjonalność polega na wykorzystaniu informacji o użytkowniku, np. w celu dobrania właściwych reklam.
- **Ransomware** - jego działanie polega na szyfrowaniu danych użytkownika celem wyłudzenia od niego opłaty za odzyskanie danych.
- **Back door** - celowo pozostawione przez twórców oprogramowania luki w zabezpieczeniach pozwalające na nieuprawniony dostęp (w tym logowanie klawiszy, ang. *keystroke logger*).
- **Logic bomb** - rodzaj luki w oprogramowaniu, która uaktywnia się pod specjalnymi warunkami.

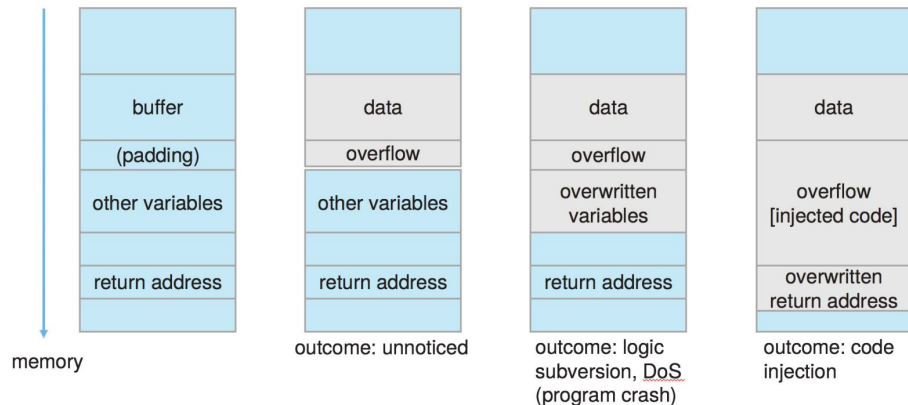


**Podstawowy sposób obrony**

**Zasada minimalnych uprawnień.**

# Wstrzykiwanie kodu

- **Wstrzykiwanie kodu** (ang. *code-injection attack*) - rodzaj ataku polegający na zmianie lub rozszerzeniu kodu uruchamialnego.
- Wstrzykiwanie kodu zwykle jest wynikiem wadliwego stosowania paradygmatów programowania w językach niskiego poziomu, np. C/C++, które umożliwiają na swobodne poruszanie się po pamięci.



Źródło: A. Silberschatz, *Operating Systems Concepts Essentials*





# Wirusy i robaki

**Wirus** - fragment kodu dołączony do programu, który potrafi sam się replikować infekując inne programy. Działanie wirusa może prowadzić w lekkiej postaci tylko do rozprzestrzeniania, a w przypadku kodu złośliwego, do niszczenia danych. Wirusy są problemem głównie dla systemów z rodziny Windows. Wirusy roznoszone są między systemami przez e-mail (w tym ataki phishingu), pobieranie zawirusowanego oprogramowania.

**Robak** - rodzaj oprogramowania “wędrującego” po sieci internet.



# Zagrożenia systemowe i sieciowe

- Model 'secure by default', czyli wszystko, co nie jest dozwolone, jest zabronione.
- **Zombie system** - opanowany przez atakującego system, z którego prowadzony jest atak.
- **Sniffing** - podsłuchiwanie.
- **Spoofing** - podszywanie.
- **Scanning** - skanowanie.



# Podstawy kryptografii

- $M$  – Tekst jawny (ang. *plaintext*, *cleartext*)
- $E()$  – Szyfrowanie (ang. *encryption*) oparty o algorytm kryptograficzny (ang. *cipher*)
- $C = E(M)$  – Kryptogram (ang. *ciphertext*)
- $M = D(C)$  – Deszyfrowanie (ang. *decryption*)  $\Rightarrow D(E(M)) = M$
- Kryptografia – nauka zajmująca się zabezpieczaniem informacji.
- Kryptoanaliza – nauka zajmująca się łamaniem kryptogramów.
- Kryptologia = Kryptografia + Kryptoanaliza.
- $H()$  – Funkcja skrótu (ang. *hash function*)
- $h = H(M)$  – Skrót (ang. *hash*)
- $S()$  – Podpisywanie cyfrowe (ang. *digital signing*)
- $V()$  – Weryfikacja podpisu (ang. *digital verification*):  $V(S(M)) = M$



# Podstawy szyfrowania

- Poufność algorytmu – jeśli bezpieczeństwo zaszyfrowanej wiadomości oparte jest o siłę algorytmu (skomplikowany algorytm).
- Poufność klucza – jeśli bezpieczeństwo zaszyfrowanej wiadomości oparte jest o siłę klucza (długość):
  - Algorytm symetryczny (np. DES – *Data Encryption Standard*):
    - $D_k(E_k(M)) = M$
  - Algorytm asymetryczny (np. RSA – *Rivest Shamir Adleman*):
    - $D_{k1}(E_{k2}(M)) = M$
    - $D_{k2}(E_{k1}(M)) = M$



# Funkcja skrótu

- Operuje na dowolnej długości wiadomości wejściowej  $M$ . Zwraca wartość hash o stałej długości  $h$ .  
$$h=H(M), \text{ gdzie } h \text{ ma długość } m$$
- Własności:
  - Łatwo obliczalna: mając  $M$  łatwo obliczyć  $h$ .
  - Jednokierunkowa: mając  $h$  trudno wyznaczyć źródłowe  $M$ :  $H(M)=h$ .
- Wolna od kolizji, ale suriekcja: istnieją takie dwie różne  $M$  i  $M'$ , że  $H(M)=H(M')$ .
- Jednoznaczna: dla każdego  $M$ :  $H(M)=H(M)$ .
- Dyfuzja, czyli cecha powodująca rozsianie bitów wiadomości jawnej w skrócie.
- Konfuzja, czyli cecha ukrywająca powiązanie pomiędzy wiadomością jawną, a skrótem.



# Podpis odręczny

- Trudny do podrobienia.
- Łatwy do zweryfikowania.
- Nieprzenoszalny na inny dokument.
- Dokumentu nie można zmienić.
- Podpisu nie można się wyprzeć.

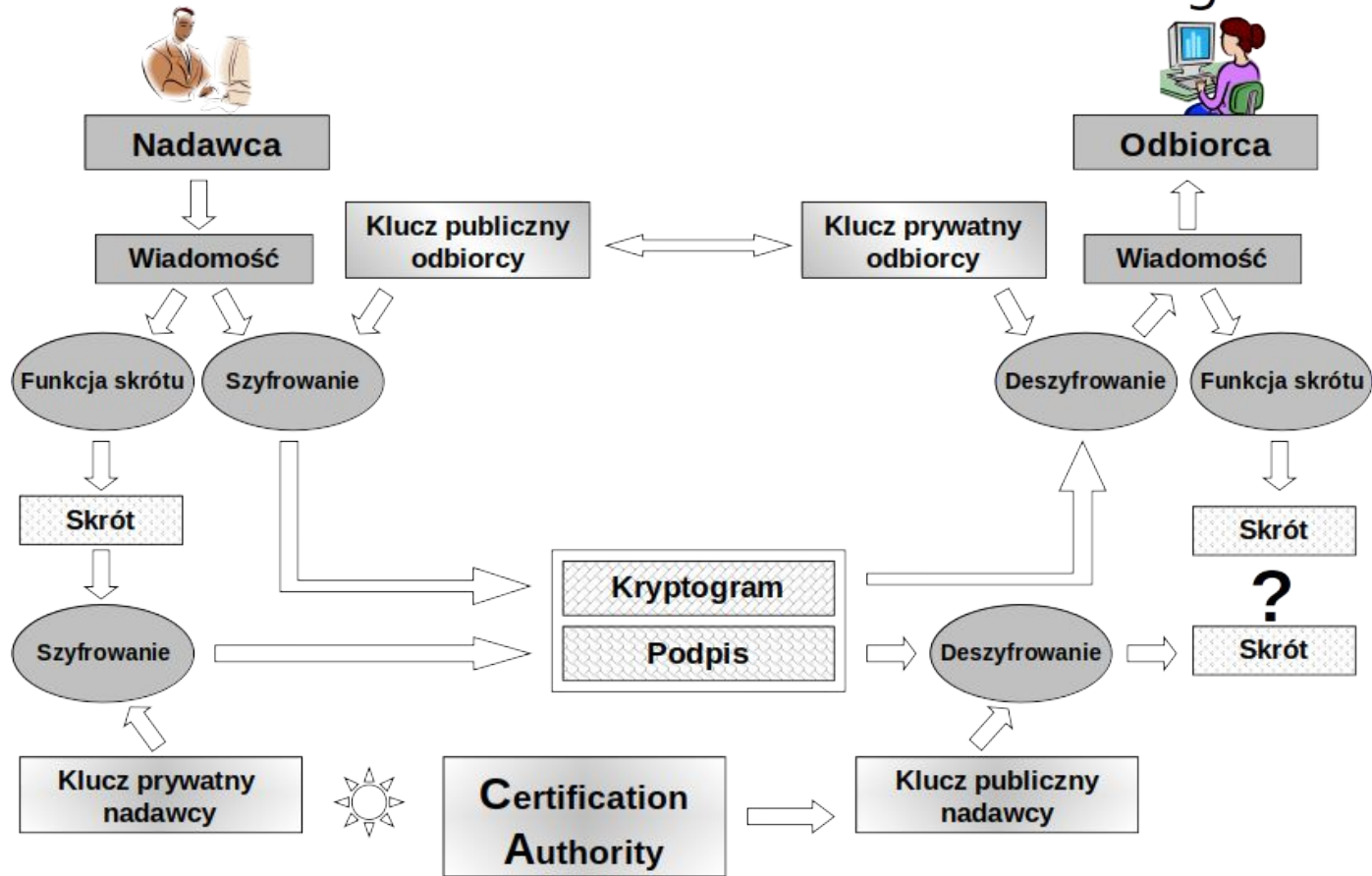
(Nie zawsze możliwe)



# Podpis cyfrowy

- $k_1$  – klucz prywatny,  $k_2$  – klucz publiczny
- $S_{k_1}(M) = E_{k_1}(H(M))$  – podpis cyfrowy  $M$
- $V_{k_2}(S_{k_1}(M), M)$  – weryfikacja podpisu  
 $D_{k_2}(S_{k_1}(M)) = ? = H(M)$   
 $D_{k_2}(E_{k_1}(H(M))) = ? = H(M)$   
 $H(M) = ? = H(M)$

# Infrastruktura Klucza Publicznego

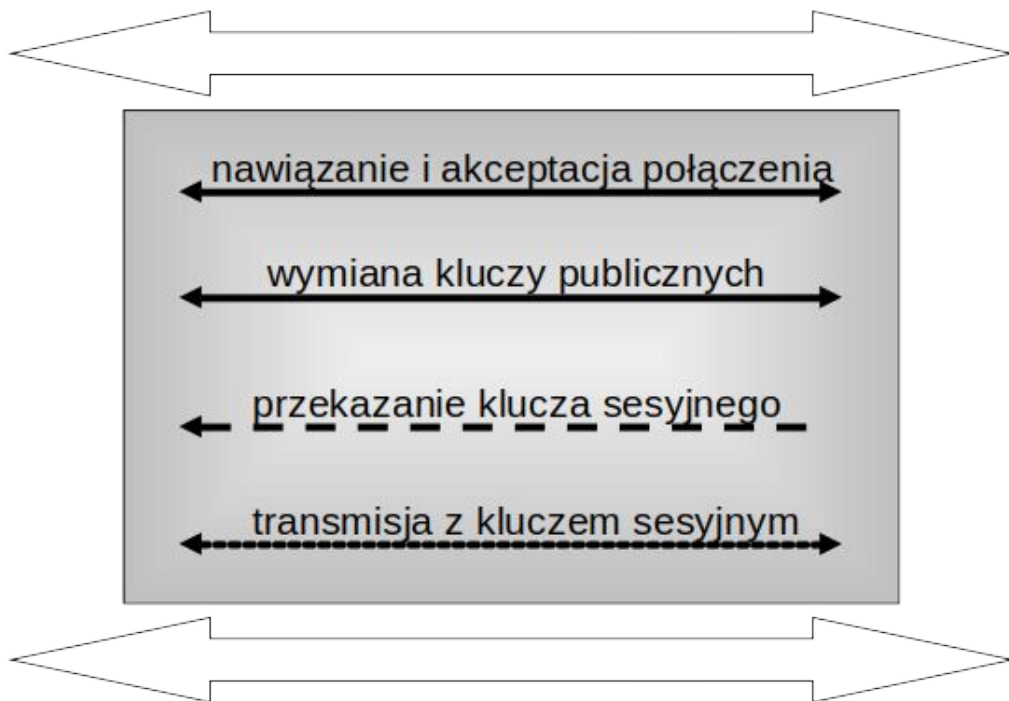




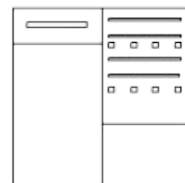
# Secure Socket Layer



Klient



HTTPS



Server

POP3S  
SMTPS



**Dziękuję za uwagę ;)**