

# Bazy Danych i Usługi Sieciowe

## Bezpieczeństwo

Paweł Daniluk

Wydział Fizyki

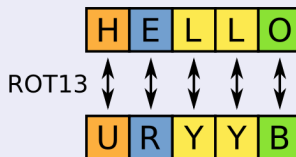
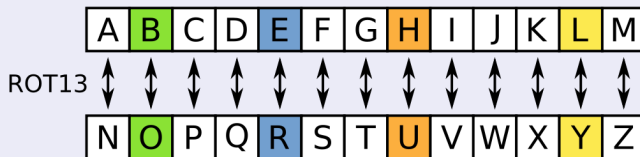
Jesień 2013



## Zabezpiecza się

- transmisje
- zasoby
- aplikacje
- maszyny

## ROT13 – szyfr podstawieniowy



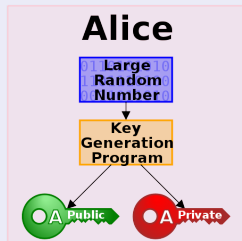
## Nowoczesne szyfry

- symetryczne (klucz prywatny) – np. DES, AES
- asymetryczne (klucz publiczny) – np. RSA

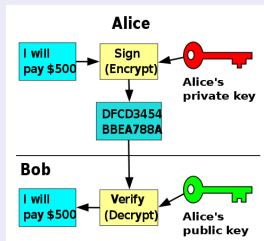
Nowoczesne szyfry są oparte na praktycznej nieodwracalności pewnych operacji arytmetycznych

# Szyfr asymetryczny

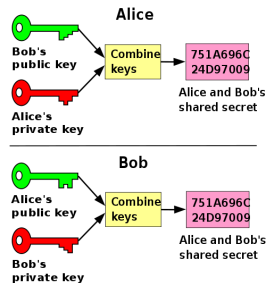
## Generowanie kluczy



## Alicja do Boba



## Wspólny klucz



Komunikat zaszyfrowany kluczem prywatnym, może zostać odczytany kluczem publicznym i odwrotnie.

Szyfry asymetryczne mogą służyć do zabezpieczania komunikatu i potwierdzania autentyczności nadawcy.

# Uwierzytelnianie i autoryzacja (authentication and authorization)

## Uwierzytelnianie

Potwierdzanie tożsamości drugiej strony (np. użytkownika).

## Można uwierzytelniać przez

- wiedzę (hasła)
- posiadanie przedmiotu (tokena, listy haseł jednorazowych, telefonu).

Hasło powinno być przesyłane zaszyfrowane, albo wcale (np. Kerberos).

# Uwierzytelnianie i autoryzacja (authentication and authorization)

## Uwierzytelnianie

Potwierdzanie tożsamości drugiej strony (np. użytkownika).

### Można uwierzytelniać przez

- wiedzę (hasła)
- posiadanie przedmiotu (tokena, listy haseł jednorazowych, telefonu).

Hasło powinno być przesyłane zaszyfrowane, albo wcale (np. Kerberos).

## Autoryzacja

Określanie uprawnień klienta (użytkownika).

ACL – Access Control Lists

# Proste uwierzytelnianie HTTP

- Od pierwszej specyfikacji HTTP 1.0
- Login i hasło wpisywane w niekonfigurowalnym oknie dialogowym wyświetlanym przez przeglądarkę
- Weryfikacja dostępu na serwerze
- Nie wymaga użycia specjalnej aplikacji (np. skryptu PHP)
- Nazwa użytkownika i hasło są przesyłane otwartym tekstem



# Proste uwierzytelnianie HTTP

- Obecnie stosuje się moduł serwera Apache
- Lista kontroli dostępu przechowywana w pliku tekstowym na serwerze (np. `.htpasswd`)
- Hasła w pliku zaszyfrowane
- Można zabezpieczyć całość lub część serwisu internetowego
- Pliki `.htaccess` - lista kontroli dostępu

# Uwierzytelnianie za pomocą formularzy

- Formularze są częścią HTML
- Dane wysyłane są na serwer za pomocą jednej z metod:
  - ▶ GET (jako parametry adresu)
  - ▶ POST (w treści zapytania (request))
- Na serwerze istnieje aplikacja (np. skrypt PHP) do której trafiają dane
- Aplikacja sprawdza poprawność danych i podejmuje decyzję o uwierzytelnieniu lub odrzuceniu
- Hasła mogą być przechowywane w bazie danych

# Ciasteczka – Cookies

- Zapewnienie “trwałości połączenia” z serwerem
- Cookie - plik tekstowy przechowujący tożsamość przeglądarki internetowej względem konkretnego serwera WWW (lub aplikacji na serwerze)
- Ustawiane wraz z pierwszym odwiedzeniem strony (lub po uwierzytelnieniu)
- Mogą zawierać dodatkowe informacje o kliencie
- Mają termin ważności (expires)

- Po uwierzytelnieniu serwer tworzy obiekt sesji
- Dane sesji w pliku lub bazie danych
- Serwer umieszcza w pliku cookie identyfikator sesji i wysyła do klienta
- W kolejnych żądaniach przeglądarka przesyła cookie do serwera, który może odnaleźć dane sesji
- Dodatkowe informacje z cookie również są przesyłane wraz z nagłówkiem żądania
- Dane z cookie nie są zabezpieczone
- Dane sesji nie są nigdy przesyłane (tylko identyfikator)

# Inne sposoby utrzymywania stanu sesji

- Klient może wyłączyć obsługę cookies
- Identyfikator sesji można przekazać jako:
  - ▶ Część adresu URL - niebezpieczeństwo przekazania identyfikatora sesji niepowołanej osobie
  - ▶ Pole ukryte formularza - w przypadku metody POST, identyfikator nie będzie widoczny w adresie URL
- Wszystkie metody narażone są na próbę przechwycenia identyfikatora sesji - jest on przesyłany otwartym tekstem

# Bezpieczne połączenie

- Protokół **SSL** - **Secure Socket Layer**
- Zapewnia uwierzytelnianie i bezpieczne przesyłanie informacji
- Obecnie standardem jest rozszerzenie protokołu SSL:
- **TLS** - **T**ransport **L**ayer **S**ecurity
- Protokół warstwy transportowej - umożliwia wykorzystanie przez protokoły z wyższych warstw (telnet, HTTP, POP3, IMAP)

# Bezpieczne połączenie

- **ClientHello** ( $K \rightarrow S$ ) dostępne parametry połączenia oraz liczba losowa używaną potem przy generowaniu kluczy
- **ServerHello** ( $S \rightarrow K$ ) wybrane parametry połączenia: wersję protokołu SSL, rodzaj szyfrowania i kompresji, oraz podobną liczbę losową
- **Certificate** ( $S \rightarrow K$ ) certyfikat pozwalający klientowi na sprawdzenie tożsamości serwera
- **ServerKeyExchange** ( $S \rightarrow K$ ) klucz publiczny
- **ServerHelloDone** ( $S \rightarrow K$ ) można przejść do następnej fazy

# Bezpieczne połączenie

- **ClientKeyExchange** ( $K \rightarrow S$ ) wstępny klucz sesji, zaszyfrowany za pomocą klucza publicznego serwera; na podstawie ustalonych w poprzednich komunikatach dwóch liczb losowych (klienta i serwera) oraz ustalonego przez klienta wstępnego klucza sesji obie strony generują klucz sesji używany do faktycznej wymiany danych.
- **ChangeCipherSpec** ( $K \rightarrow S$ ) można przełączyć się na komunikację szyfrowaną
- **Finished** ( $K \rightarrow S$ ) już zaszyfrowany, gotowość do wymiany danych
- **ChangeCipherSpec** ( $S \rightarrow K$ ) potwierdzenie przejścia na komunikację szyfrowaną
- **Finished** ( $S \rightarrow K$ ) już zaszyfrowany, gotowość do wymiany danych

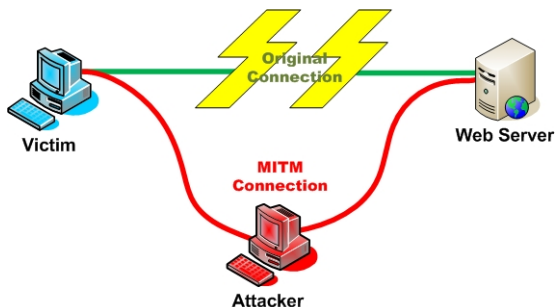


# Certyfikaty

- Certyfikat zawiera nazwę i klucz publiczny właściciela.
- Certyfikat jest podpisany przez wystawcę (CA – *Certification Authority*)
- CA tworzą strukturę hierarchiczną.
- Certyfikaty głównych CA są dołączone do przeglądarek.

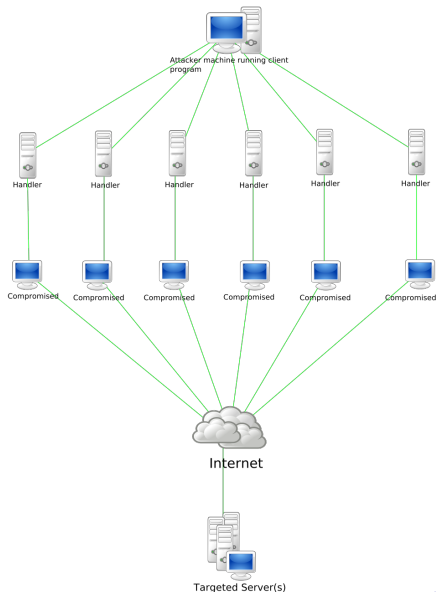
Przejęcie CA powoduje załamanie systemu.

# Man-in-the-middle attack



- Aspidistra (1945)
- Turing porn farm
- Kradzież informacji (np. kluczy i haseł)

# Distributed Denial of Service (DDoS)



# Distributed Denial of Service (DDoS) c.d.

## Mechanizmy

- ICMP –
- SYN – nawiązywanie i porzucanie połączeń TCP
- Na poziomie aplikacji
- RUDY – R-U-Dead-Yet? – W nieskończoność otwarte żądanie POST

## Strategie

- atak rozproszony
- atak odbity
- sieci botów (botnets)
- atak niezamierzony (Slashdotting)

## Obrona

- firewall
- dedykowane urządzenia i aplikacje

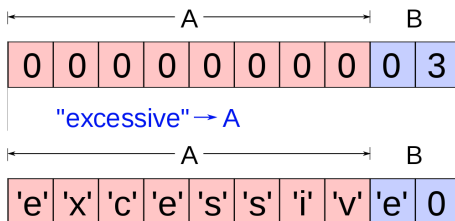
# Atak na program

- Wojny rdzeniowe (Tron ;-)
- Wirusy
  - ▶ 1949 – “Theory of self-reproducing automata” John von Neumann
  - ▶ 1971 – Creeper – ARPANET
  - ▶ 1981 – Elk Cloner – Apple DOS 3.3
  - ▶ 1986 – (c)Brain
- Makro-wirusy

## Wektory infekcji

- pliki wykonywalne
- systemowe przestrzenie dysków i dyskietek (boot records)
- skrypty
- dokumenty zawierające makra
- dowolne pliki powodujące wystąpienie błędów w programach, które je otwierają

# Przepełnienie bufora (*buffer overflow*)



Pozwala zmienić działanie programu (np. przejąć nad nim kontrolę).

## Przeciwdziałanie

- kontrola wyjścia poza zakres tablicy
- bezpieczne biblioteki
- randomizacja przestrzeni adresowej

# Cross Site Scripting (XSS)

Zawartość wprowadzona przez użytkownika zostaje pokazana bez kontroli.

- tymczasowe
- stałe

## Przykład

- 1 Alicja często odwiedza stronę Boba. Strona Boba wymaga logowania i przechowuje prywatne dane.
- 2 Mallory zauważa, że strona Boba jest podatna na XSS.
- 3 Mallory tworzy URL wykorzystujący lukę i wysyła Alicji e-mail z linkiem, który wskazuje na stronę Boba i zawiera kod umożliwiający przejęcie hasła.
- 4 Alicja klika na link, będąc zalogowana na stronie Boba.
- 5 Skrypt Mallory'ego wykonuje się w przeglądarce Alicji, tak jakby pochodził z serwera Boba i wysyła szczegóły sesji Mallory'emu.

# SQL injection





## Aplikacji

- poprawne programowanie
- sanityzacja napisów
- separacja programów
- wirtualizacja

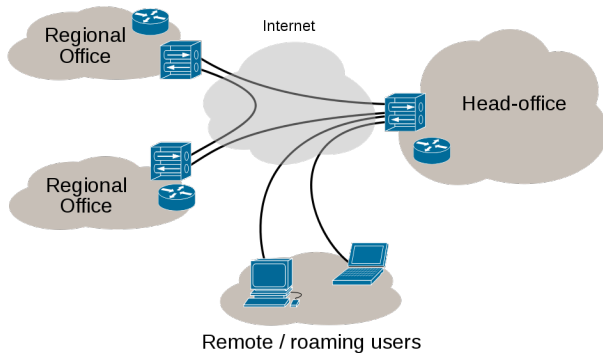
## Sieci

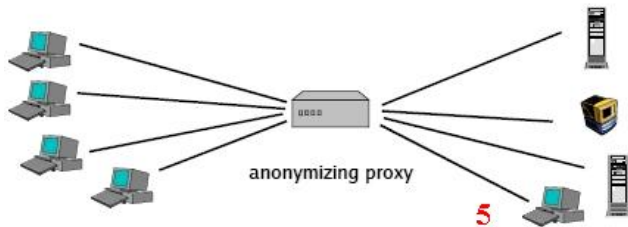
- blackholes
- honeypots (atak przez fałszywe domeny)

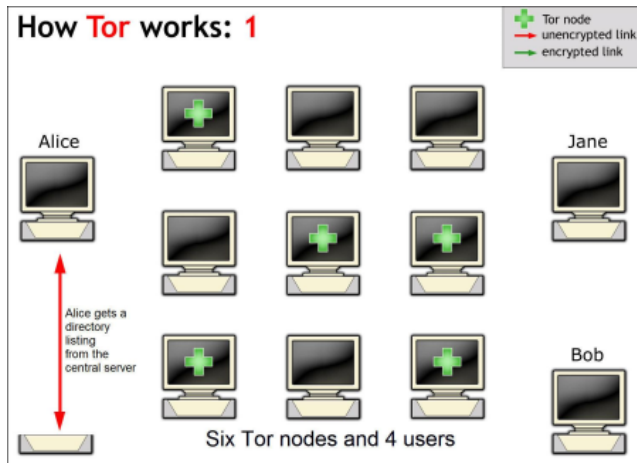
Good design vs. Security through obscurity

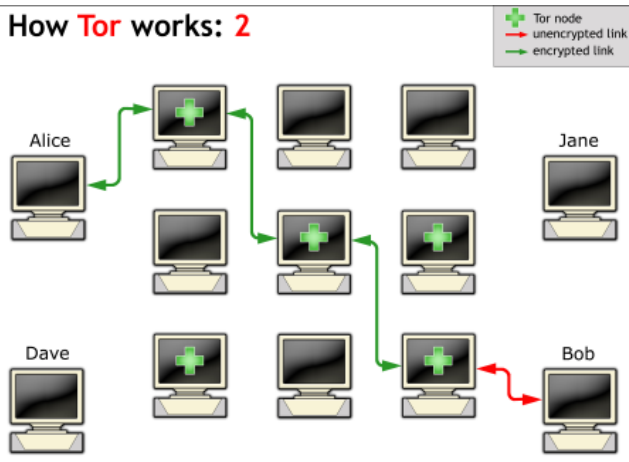
# Tunele i VPN (*Virtual Private Networks*)

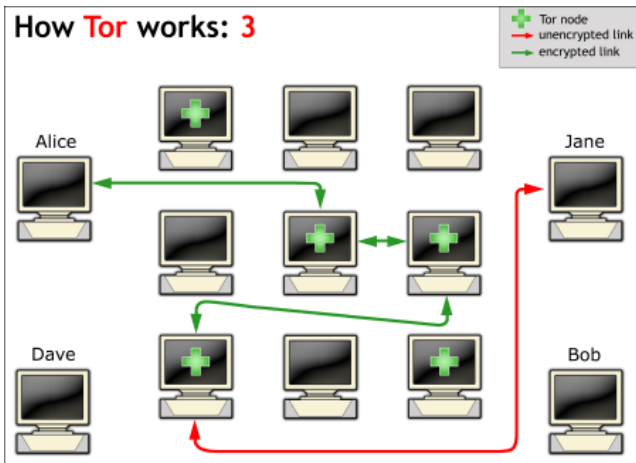
## Internet VPN

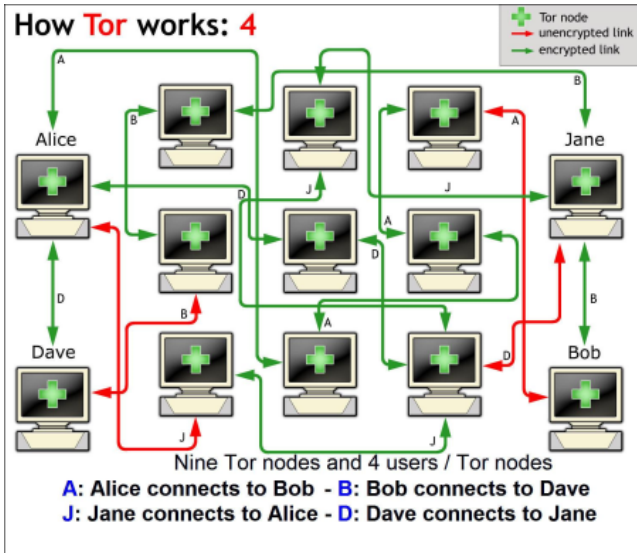












[http://bioexploratorium.pl/wiki/  
Bazy\\_danych\\_i\\_uslugi\\_sieciowe\\_-\\_2013z](http://bioexploratorium.pl/wiki/Bazy_danych_i_uslugi_sieciowe_-_2013z)