Podstawy kryptografii

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. zimowy 2025/2026

inf.ug.edu.pl/~amb/

Uwierzytelnianie

Metody uwierzytelniania

- Cztery metody
  – wiedza – hasło, PIN, zawołanie-odzew, protokoły wiedzy zerowej, ...
  – posiadanie przedmiotu – token, karta, klucz (fizyczny przedmiot) ..
  – cechy indywidualne i niepowtarzalne – wzorzec siatkówki oka, odcisk palca, kształt twarzy ...
  – jw. ale dotyczące zachowań – podpis odręczny, głos, sposób chodzenia, pisania ...
- Problemy
  – wiedza – można nie zauważyć kompromitacji
  – posiadanie przedmiotu – nie ma szans na backup
  – cechy indywidualne i niepowtarzalne – nie mogą być zmienione

Metody uwierzytelniania – hasło

- Hasło
  – jest na ogół przesyłane jawnie, tylko lokalne zastosowanie
- Ataki na hasło i obrona przed nimi
  - Kradzież: podglądanie, kopie papierowe, wirus przechwytujący hasła
    – unikanie sytuacji, programy antywirusowe
  - Podstęp (phishing)
    – ostrożność (edukacja użytkowników)
  - Hasła dawne mogą być skompromitowane
    – okresowa zmiana hasła (czy wolno wracać do dawnych?)
    – obowiązek haseł jednorazowych
  - Hasła domyślne – natychmiastowa zmiana hasła
  - Przechowywanie haseł
    – kradzież również przez admina
    – serwer przechowuje jedynie skrót hasła i porównuje ze skrótem hasła podanego

Ataki na hasło i obrona c.d.

- Zgadywanie hasła
  - brutalne (wszystkie hasła)
    – postać syntaktyczna: długość, ≥ 8, lepiej 12 znaków
  - słownikowe (użytkownicy częściej wybiorą hasła z małego podzbioru możliwych)
    – postać syntaktyczna: kilka rodzajów znaków
    – testowe łamanie hasła (mało skuteczne)
    – serwer nie pozwala wykonać zbyt wielu prób
  - *blue tables*, przygotowane skróty haseł ze słownika, po kradzieży skrótów haseł szukanie elementu
    w $\{h(p)|p \in$ hasła użytkowników$\} \cap \{h(w)|w \in$ słownik$\}$
    – solenie (salted) haseł: $\langle s, h(p||s)\rangle$, s losowe ziarno uniemożliwia użycie błękitnych tablic

Dodatkowe zasady uwierzytelniania

- Single sign-on uwierzytelnianie w systemie następuje raz
  – zmniejsza szanse na przechwycenie, zachęca do trudniejszych haseł
  – jeden serwer uwierzytelniający w systemie
- Hasła jednorazowe
  – są przekazywane innym kanałem (np. sms)
  – lista skrótów $h_n(s), n = 0, 1, \ldots, 100$, przechowywanie wartości licznika i odpowiedniego skrótu
  – funkcja generująca hasło $G(X, p, t)$ w zależności od użytkownika X, hasła p oraz znacznika czasu t
  – zawołanie–odzew, żądający wysyła zawołanie x, uwierzytelniający się potrafi wyprodukować odzew $G(x, p)$
  – de facto jest to podpis (ślepy) na żądanie
  – często dedykowane urządzenia (token)

Klucze

Klucze – długość

- „Moc" szyfrowania powinna zależeć tylko od długości klucza
- Moc przestrzeni kluczy: najczęściej 2^n, klucze n-bitowe
- Atak brutalny – trzeba przeszukać wszystkie klucze
  – możliwość wykorzystania równoległości (specjalna architektura, sieć komputerów)
  – trzeba mieć informację, że klucz został znaleziony
  – typowy problem dla kryptografii symetrycznej
  – 56 bitów (DES) dziś to za mało, używa się kluczy ≥ 100 bitów
- Kryptografia asymetryczna – problemy matematyczne o pewnej złożoności
  – RSA lub logarytm dyskretny: złożoność $\approx exp(n^{\frac{1}{3}})$
  – dziś używa się kluczy długości co najmniej 1024 bitów

Klucze – długość, c.d.

- Kryptografia krzywych eliptycznych – ok. trzykrotnie krótsze klucze niż w RSA
- Nowe, nieprawdopodobne techniki:
  – DNA:„obliczenia” biochemiczne
  – komputer kwantowy: znika wykładnicza złożoność przy faktoryzacji, RSA staje się bezużyteczne
    - przez długi czas szczytowym osiągnięciem: $15 = 3 \cdot 5$
    - dziś również $56153 = 241 \cdot 233$
    - nie ma dowodu, że technika jest skalowalna, nie widać by faktoryzacja działała w praktyce
    - kryptografia postkwantowa – „dla większości systemów nie są znane żadne zastosowania obliczeń kwantowych w ich kryptoanalizie”

Klucze – jakość

- Bajt $\neq$ 8 bitów
  – ascii = 95 znaków
  – litery i cyfry = 62 znaki
- Atak słownikowy
  – dane użytkownika, imiona, nazwy geograficzne, łatwe kombinacje znaków na klawiaturze
  – warianty z liczbami, znakami specjalnymi
  – słownik duży ma 100 tys słów, wybór słowa to ok. 17 bitów
  – litera w słowie angielskim ≈ 1.3 bita
- Przeciwdziałanie:
  – dłuższe hasła, pary słów, passphrase zamiast password
- ale niektóre implementacje mogą obcinać dłuższe hasła
  – automatyczne dodawanie losowego ziarna do haseł
- Klucze losowe: generator liczb losowych

Problem uzgodnienia/dystrybucji kluczy

- Dystrybucja – przekazanie klucza drugiej stronie
  – obie strony mają wspólny klucz
  – ale podczas przekazywania klucz może być podsłuchany
- Uzgodnienie – po wykonaniu protokołu obie strony mają wspólny klucz
  – możliwe dopiero w kryptografii asymetrycznej (DH)
- Kryptografia symetryczna
  – musi być przekazanie klucza
  – odrębny klucz dla każdej pary uczestników
- Kryptografia asymetryczna
  – radykalnie mniejsza wydajność
  – problem uwierzytelniania

Protokoły kryptograficzne

Protokoły kryptograficzne

- Protokół jest wykonywany przez kilku uczestników
  – algorytm ma tylko jednego wykonawcę
- Protokół jest zaprojektowany dla pewnych wykonawców
  – ale trzeba analizować możliwość wykonania przez niezaproszonych gości
- Główny przykład: man-in-the-middle, atak ze środka systemu
- Idea: Mariola gra w szachy z dwoma arcymistrzami
  – na pewno nie przegra z oboma
  – ponieważ tylko przenosi ruchy z szachownic
- Atakujący może wykonywać ten sam protokół z różnymi uczestnikami, w sumie wykonań jest więcej niż jedno
  – może też atakować z przesunięciem w czasie

Protokół żaby o szerokich ustach (wide mouth frog protocol)

- Istnieje zaufany pośrednik Tadeusz, uzgodnił on klucze tajne ze wszystkimi uczestnikami
- Alicja⇒ Tadeusz: $id(A), E_A(T_A, id(B), K)$ gdzie T_A jest znacznikiem czasu, K jest kluczem sesyjnym dla sesji z Bolkiem
  – Tadeusz⇒ Bolek: $E_B(T_B, id(A), K)$, T_B jest nowym znacznikiem czasu
  – i Tadeusz i Bolek oceniają czy znacznik czasu jest sensowny
- duże wymagania od Tadeusza (Bolek może nie być dostępny)
  – duże wymagania od uczestników (Alicja może nie mieć kompetencji do zaproponowania klucza sesyjnego, każdy z uczestników może mieć niezbyt dokładny zegar)

Atak na protokół żaby o szerokich ustach

- Mariola zapisuje komunikat Tadeusza do Bolka $E_B(T_B, id(A), K)$
  – po chwili przesyła go z powrotem do Tadeusza
  – wygląda jak żądanie Bolka komunikacji z Alicją
  – Tadeusz przesyła do Alicji komunikat z nowym znacznikiem czasu
- Wielokrotne powtórzenie tej operacji powoduje, że klucz jest ważny bardzo długo
  – być może ułatwi to atak Marioli
  – atak byłby wykryty, gdyby uczestnicy zauważyli, że klucz jest ciągle ten sam
  – ale przechowywanie bazy dawnych kluczy jest niewskazane z wielu powodów
- Atak jest możliwy ponieważ Tadeusz zmienia znacznik czasu

Obrona przed atakiem przez powtórzenie

- Mariola może zachować podsłuchane wiadomości i powtórzyć je po pewnym czasie
- Obrona: uniemożliwienie powtórzeń
  – numery kolejne: wiadomości są numerowane, uczestnicy pamiętają jaki numer był ostatni, nie dopuszczą, by się powtórzył
  – możliwość takiej numeracji nie jest oczywista (nowi uczestnicy, ryzyko przekłamania na linii)
  – znaczniki czasu: wiadomości mają krótki okres ważności
  – wymaga to dokładnych zegarów (synchronizacja)
  – liczby losowe (jednorazowe): Alicja generuje liczbę losową i żąda by Bolek odesłał jej tę liczbę

Atak odmowy usługi (DOS – denial of service)

- Wersja ataku dla serwera uwierzytelniającego
  – Alicja żąda od Tadeusza przesłania komunikatu do Bolka
  – np. protokół żaby o szerokich ustach
  – Bolek może nie odpowiadać, może nie istnieć
  – zamiarem Alicji może być obciążenie zadaniami Tadeusza, a nie komunikacja z Bolkiem
- Przeciwdziałanie atakowi
  – Tadeusz nie komunikuje się z Bolkiem
  – Tadeusz przygotowuje bilet dla Bolka, ale wręcza go Alicji
  – Alicja jest odpowiedzialna za komunikację z Bolkiem
  – praca Tadeusza nie przekracza znacząco pracy Alicji

Protokół Needhama-Schrödera – bilety jako obrona przed DOS

- $A \Rightarrow T : id(A), id(B), r_A$
- $T \Rightarrow A : E_A(r_A, id(B), K, E_B(K, id(A)))$ – klucz plus bilet dla Bolka
- Alicja już zna klucz i przesyła bilet Bolkowi
  – $A \Rightarrow B : E_B(K, id(A))$
- Gdyby tu zakończyć, to Mariola mogłaby wielokrotnie przesyłać ten bilet
  – Bolek wielokrotnie szyfrowałby wiadomości tym kluczem ułatwiając kryptoanalizę
- W protokole są dodatkowe kroki
  – $B \Rightarrow A : E_K(r_B)$
  – $A \Rightarrow B : E_K(r_B - 1)$
  – Bolek nie użyje klucza dopóki nie ma pewności, że druga strona potrafi go użyć, Mariola na razie nie potrafi
  – szyfrowanie liczb losowych nie ułatwia kryptoanalizy

Kerberos

- Opracowany w MIT
  – używa kryptografii symetrycznej
  – do uwierzytelniania użytkowników w dużej, ale scentralizowanej, sieci komputerowej
  – głównie asymetrycznie: klient uwierzytelnia się przed serwerem (dostęp do konta, drukarki, innych zasobów)
- Założenia: co najmniej jeden centralny serwer
  – uwierzytelniający użytkowników (hasła, itp.)
  – być może inny/dalsze serwery generujące bilety
  – którymi użytkownicy będą się posługiwać żądając dostępu do usług
  – uwierzytelnienia mają ograniczony czas ważności

Kerberos, protokół

- *Alicja* $\Rightarrow$ *Tadeusz*: zgłoszenie chęci kontaktu z Bolkiem
- *Tadeusz* $\Rightarrow$ *Alicja* : $E[K_A](K_{AG})$ – zaszyfrowany kluczem Alicji klucz sesyjny dla Grażyny (serwer biletujący) K_{AG} oraz
  – $\langle id(A), E[K_G](id(A), t_1, K_{AG})\rangle$ – bilet dla Grażyny ze znacznikiem czasu t_1
- *Alicja* $\Rightarrow$ *Grazyna*: powyższy bilet oraz
  – $E[K_{AG}](id(A), t_2)$
- Grażyna z biletu dowiaduje się, że rozmawia z Alicją
  – sprawdza, że otrzymany drugi znacznik czasu jest dobry
  – generuje bilet dla Alicji do przekazania Bolkowi
  – bilet jest ważny na czas określony
- *Alicja* $\Rightarrow$ *Bolek*: bilet od Grażyny z kolejnym znacznikiem czasu

Kerberos, cechy

- Hierarchia
  – jeden serwer uwierzytelniający uczestników
  – w szczególności znający klucze tajne
  – wszystkie inne serwery wierzą w to uwierzytelnienie
- Bilety do poszczególnych usług przydzielane są przez serwery biletujące
  – serwer przydzielający bilety wierzy głównemu serwerowi
  – serwery udostępniające usługi wierzą odpowiednim serwerom
- Uczestnik jest uwierzytelniany wobec usług
  – ale nie ma symetrii
  – zawsze inicjatywa komunikacji i konieczność uwierzytelnienia się jest po stronie uczestnika

SSL (secure socket layer) oraz TLS (transport layer security)

- Używane do komunikacji„zwykłego” użytkownika z serwerem hhtp, serwer powinien być uwierzytelniony, uczestnik niekoniecznie
  – opracowane przez Netscape, 1994
  – oraz Internet Eng. Task Force, 1999
- Alicja nawiązuje kontakt z Bolkiem (serwer http)
  – hello: uzgodnienie algorytmów z listy preferowanych
  – wysyła również (jawnie) znacznik czasu oraz liczbę losową
  – *Bolek* $\Rightarrow$ *Alicja*: certyfikat i też znacznik czasu i liczba losowa
  – *Alicja* $\Rightarrow$ *Bolek*: zaszyfrowany klucz wstępny
  – klucze do komunikacji są generowane z klucza wstępnego i z podanych znaczników czasu i liczb jednorazowych
- Użycie znaczników czasu i liczb jednorazowych uniemożliwiają atak przez powtórzenie

SSL/TLS c.d.

- Komunikacja używa 6 kluczy (szyfrowanie, MAC, szyfr blokowy, osobno w każdą stronę komunikacji)
  – *Alicja* $\Rightarrow$ *Bolek*: skompresowana i zaszyfrowana wiadomość (klucz do szyfrowania i do trybu CBC blokowego) plus MAC
  – *Bolek* $\Rightarrow$ *Alicja*: analogicznie