Kryptografia i bezpieczeństwo systemów informatycznych

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. letni 2025/2026

inf.ug.edu.pl/~amb/

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

Schemat uzgodnienia klucza Diffie-Hellmana

- uzgodnione: grupa G (np. $\mathbb{Z}_p^*$ – p liczba pierwsza) i generator γ
  – uczestnicy mają klucze prywatne (Alicja a, Bolek b)
  – znane są ich klucze publiczne
- Alicja $A = \gamma^a \mod p$, Bolek $B = \gamma^b \mod p$
  – kluczem wspólnym dla Alicji i Bolka jest $\gamma^{a \cdot b} = A^b = B^a \mod p$
- każdy z uczestników może go obliczyć z dostępnej informacji
  – dla intruza znalezienie wspólnego klucza jest równoważne z obliczeniowym problemem DH
  – sprawdzenie kandydata na klucz jest decyzyjnym problemem DH
  – też podejrzewanym, że jest trudny

Atak na schemat Diffie-Hellmana

- Ustalono liczbę pierwszą p oraz generator γ
  – uczestnicy mają klucze prywatne (Alicja a, Bolek b)
- Przesyłają sobie swoje klucze publiczne: Alicja $A = \gamma^a \mod p$, Bolek $B = \gamma^b \mod p$
  – Mariola zatrzymuje obie przesyłki, wybiera swój klucz m i przesyła $\gamma^m \mod p$ do Alicji i Bolka
  – Alicja oblicza klucz wspólny $\gamma^{m \cdot a} \mod p$, a Bolek $\gamma^{m \cdot b} \mod p$
  – oba klucze zna Mariola
  – Alicja i Bolek sądzą, że posługują się wspólnym kluczem
  – tymczasem korespondencja przechodzi przez Mariolę, która na bieżąco odszyfrowuje wiadomości jednym kluczem i szyfruje drugim

Obrona przed atakiem (protokół STS – station-to-station)

- Zaufany pośrednik potwierdzający tożsamość uczestników
  – Alicja wysyła klucz publiczny $A = \gamma^a$
  – Bolek oblicza wspólny klucz $K = \gamma^{a \cdot b}$, wysyła Alicji klucz publiczny γ^b, oraz zaszyfrowane i podpisane oba klucze $E[K]sigB(\gamma^b, \gamma^a)$
  – Alicja oblicza wspólny klucz K, odszyfrowuje podpisaną wiadomość i prosi Tadeusza o weryfikację podpisu Bolka
  – potem wysyła Bolkowi zaszyfrowane i podpisane przez siebie oba klucze $E[K]sigA(\gamma^b, \gamma^a)$
  – teraz Bolek prosi Tadeusza o weryfikację podpisu
- Mariola nie może wejść w rolę żadnego z uczestników, bo Tadeusz nie potwierdzi, że jej podpis jest podpisem Bolka czy Alicji

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 ten sam
  – ale przechowywanie bazy dawnych kluczy jest niewskazane
- Atak jest możliwy ponieważ Tadeusz zmienia znacznik czasu

Obrona przed atakiem przez powtórzenie

- Atak: Mariola może zachować podsłuchane wiadomości i powtórzyć je po pewnym czasie
- Obrona: unieważnienie 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 być niedostępny, może nawet nie istnieć
  – zamiarem Alicji może nie być komunikacja z Bolkiem a obciążenie zadaniami Tadeusza
- Przeciwdziałanie atakowi
  – Tadeusz nie komunikuje się z Bolkiem
  – Tadeusz przygotowuje bilet dla Bolka ale wręcza go Alicji
  – np. protokół Needhama-Schrödera
  – 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

Atak na schemat N-Sch

- Mariola zachowuje przesyłane komunikaty
  – załóżmy, że znalazła dawny klucz sesyjny K
  – zaczyna od środka $M \Rightarrow B : E_B(K, id(A))$
  – Bolek jest przekonany, że rozmawia z Alicją i kontynuuje wymianę informacji
- ochrona: używanie znaczników czasu
  – klucz sesyjny jest ważny tylko krótko
  – obie strony muszą mieć wiarygodne zegary
  – nadal możliwy jest atak, ale w wąskim przedziale czasowym

Standardowe protokoły kryptograficzne

Kerberos

- Opracowany w MIT
  – używa kryptografii symetrycznej
  – do uwierzytelniania użytkowników w dużej, ale scentralizowanej, sieci komputerowej
  – 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 v.5

1. *Alicja* $\Rightarrow$ *Tadeusz*: zgłoszenie chęci kontaktu z Bolkiem, liczba losowa r_A
2. *Tadeusz* $\Rightarrow$ *Alicja*: klucz ze znacznikiem czasu oraz bilet dla Bolka (niezaszyfrowany) $E_A(r_A, id(B), K, t)$, $E_B(K, id(A), t)$
3. Alicja poznaje klucz K, określa swój czas t' mieszczący się w terminie ważności wymiany klucza
   – *Alicja* $\Rightarrow$ *Bolek*: $E_B(K, id(A), t)$, $E_K(id(A), t')$
4. Bolek poznaje klucz K, poznaje znacznik Alicji t', udowadnia, że potrafił wszystko odszyfrować
   – *Bolek* $\Rightarrow$ *Alicja*: $E_K(id(B), t' + 1)$

- oparty o protokół N-Sch. ale z dodanymi znacznikami czasu
- nie ma konieczności powtórnego szyfrowania biletów
- żąda się znajomości klucza, co nie zawsze jest słuszne
- szyfrowanie jest jednocześnie uwierzytelnieniem, dziś uważa się, że każda z funkcji powinna służyć jednemu celowi

Kerberos, protokół pełniejszy

- *Alicja* $\Rightarrow$ *Tadeusz*: zgłoszenie chęci kontaktu z Bolkiem
- *Tadeusz* $\Rightarrow$ *Alicja* : $E_A(K_{AG})$ – zaszyfrowany kluczem Alicji klucz sesyjny dla Grażyny (serwer biletujący) K_{AG} oraz
  – $\langle id(A), E_G(id(A), t_1, K_{AG})\rangle$ – 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
- Konieczność posiadania wiarygodnych zegarów przez wszystkich uczestników
- 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 również 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
- Alicja ma pewność, że rozmawia z Bolkiem
- Bolek nie ma żadnej wiedzy z kim rozmawia, jeśli zachodzi taka potrzeba, konieczne jest odrębne uwierzytelnienie

SET (secure electronic transaction)

- Standard opracowany przez Visa i MasterCard (1996)
- Jest bardzo złożony, zawiera m.in. SSL
- Założenia: Alicja kupuje towar w sklepie, płaci kartą banku
  – sklep nie może się dowiedzieć szczegółów karty
  – bank nie może wiedzieć, co Alicja kupuje
  – Alicja, Bank i Sklep mają swoje klucze publiczne i prywatne, wspólna jest funkcja skrótu
- Alicja generuje dwa dokumenty:
  – Zamówienie (dane o Alicji i banku oraz o towarach)
  – Płatność (dane o Alicji, karcie oraz o sklepie)
  – oblicza: $h_1 = H(E_S(Zam))$, $h_2 = H(E_B(Pla))$, $H(h_1, h_2)$
  – podpisuje połączony skrót
  – przesyła do sklepu zaszyfrowane dokumenty i podpis

SET c.d.

- Sklep odszyfrowuje zamówienie
  – weryfikuje podpis Alicji pod zaszyfrowanymi dokumentami
  – wie, że jest podpis pod tym zamówieniem
  – przesyła do banku $H(E_S(Zam))$, $E_B(Pla)$ i podpis Alicji
  – nie potrafi skutecznie połączyć innego zamówienia z inną płatnością
- Bank odszyfrowuje dokument płatności
  – weryfikuje podpis Alicji pod zaszyfrowanymi dokumentami
  – wie, że jest podpis pod tym dokumentem płatności
  – przesyła do sklepu gwarancję, że zapłata będzie dokonana
- Wszystkie klucze publiczne wymagają dodatkowych narzędzi jak zwykle