Bezpieczeństwo w inżynierii oprogramowania

Mechanizmy uwierzytelniania

Arkadiusz Chrobot

Katedra Systemów Informatycznych

19 października 2024

Plan

1. Wstęp
2. Definicje
3. Metody uwierzytelniania
   - Metody oparte na wiedzy
   - Metody oparte na posiadaniu
   - Metody oparte na cechach/atrybutach
   - Metody oparte na lokalizacji
   - Metody oparte na zachowaniu

Wstęp

Uwierzytelnianie (ang. *authentication*) jest jedną z trzech składowych *złotego standardu* bezpieczeństwa (uwierzytelnienie, autoryzacja, audyt), będącego podstawą zabezpieczeń każdego systemu informatycznego. Z tego względu temat ten jest poruszany w ramach wykładu dotyczącego bezpieczeństwa w inżynierii oprogramowania. Omówione zostaną wybrane metody sprawdzania tożsamości użytkowników. Na początku jednak zostaną przedstawione definicje podstawowych pojęć związanych z tą tematyką. Formalnie pojęcia związane z usługami bezpieczeństwa (ang. *security services*) i pokrewnymi mechanizmami określa norma ▸ ISO-7498-2, jednak została ona opracowana stosunkowo dawno (rok wydania 1989). W związku z tym nowsze definicje nowszych terminów będą pochodziły z innych ▸ źródeł, między innymi takich, jak normy ▸ NIST.

Definicje

Identyfikacja

To proces, w którym użytkownik przedstawia systemowi informatycznemu (oprogramowaniu, usłudze) swoją tożsamość.

Uwierzytelnienie

To proces, w którym użytkownik *potwierdza* swoją tożsamość, *udowadnia*, że jest tym za kogo lub co się podaje. Słowo *użytkownik* może oznaczać zarówno człowieka, jak i urządzenie. Uwierzytelnienie może dotyczyć również *pochodzenia danych* (ang. *data origin*). Oba procesy (identyfikacja i uwierzytelnienie) mogą zachodzić jednocześnie (wprowadzenie loginu i hasła na tej samej formatce, biometria) [1].

Definicje

Silne uwierzytelnienie

Nie ma ▸ formalnej definicji silnego uwierzytelnienia (ang. *strong authentication*). Pierwotnie ten termin był utożsamiany z uwierzytelnieniem dwu lub wieloskładnikowym. Współcześnie jednak wymagane jest, aby metoda potwierdzania tożsamości spełniająca definicję uwierzytelniania silnego była ▸ odporna na określone typy ataków. Reasumując, można przyjąć, że silne uwierzytelnianie znacząco redukuje możliwości przejęcia tożsamości uprawnionego użytkownika przez intruzów, ale również ogranicza sposoby zaprzeczenia, że ta tożsamość należy do tego użytkownika.

Uwierzytelnienie jednoskładnikowe

Proces uwierzytelnienia, w którym tożsamość użytkownika jest potwierdzana przy pomocy tylko jednej metody, najczęściej bazującej na jego wiedzy.

Definicje

Uwierzytelnienie wieloczynnikowe

Uwierzytelnienie wieloczynnikowe (ang. *multi-factor authentication — MFA*) polega na jednoczesnym zastosowaniu kilku metod uwierzytelniania, bazujących najczęściej na wiedzy, stanie posiadania lub cechach/atrybutach użytkownika. **Ważnym jest, aby wszystkie składniki nie należały do tej samej kategorii metod, np. opartych tylko na wiedzy użytkownika!** Najczęściej spotykanym rozwiązaniem z tej kategorii jest uwierzytelnianie dwuskładnikowe (ang. *two-factor authentication — 2FA*).

Autoryzacja

To proces decydowania do jakich zasobów systemu informatycznego może uzyskać dostęp użytkownik, *który pomyślnie został uwierzytelniony.*

Definicje

Kontrola dostępu

Identyfikacja, uwierzytelnienie i autoryzacja są częścią procesu *kontroli dostępu* (ang. *access control*).

Metody uwierzytelniania

Istnieje kilka klasyfikacji metod uwierzytelniania [1]. Najczęściej używana i prawdopodobnie najistotniejsza jest ta, która klasyfikuje te metody względem tego, co jest w nich dowodem tożsamości użytkownika. Tę klasyfikacja zostanie przedstawiona w dalszej części wykładu.

Metody oparte na wiedzy

W metodach opartych na wiedzy dowodem potwierdzającym tożsamość użytkownika jest sekret znany tylko jemu i systemowi przeprowadzającemu uwierzytelnienie. Dlatego te metody opisywane są też zdaniem *to co użytkownik wie*. Do tej kategorii zalicza się hasła statyczne, PIN, wzory rysowane na ekranie dotykowym, pytania pomocnicze, itp. W tego typu metodach sekret nie powinien być przechowywany w systemie uwierzytelniania w *formie jawnej* (ang. *plain text*). W przypadku haseł statycznych przechowuje się jedynie ich skróty, wygenerowane za pomocą takich algorytmów jak *scrypt*, *bcrypt*, *PBKDF2* i *Argon2*. Wybór algorytmu zależy najczęściej od wymogów standardów, którym podlega oprogramowania z danej dziedziny oraz od zasobów platformy sprzętowej, na której będzie wykonywane uwierzytelnienie.

Metody oparte na wiedzy

Oprócz hasła argumentem wejściowym tych algorytmów jest *sól* (ang. *salt*), czyli losowa, unikatowa wartość, która jest zapisywana jest też w skrócie. Dzięki niej, jeśli nawet dwóch lub większa liczba użytkowników wybierze przez przypadek to samo hasło, to skróty tych haseł będą inne. Dodatkowo stosowany jest również *pieprz* (ang. *pepper*). To również losowa wartość, ale przechowywana w osobnym miejscu, np. w *sprzętowym module bezpieczeństwa* (ang. *Hardware Security Module — HSM*), która zazwyczaj jest dodawana do każdego hasła przed obliczeniem jego skrótu. W niektórych systemach uwierzytelniania pieprz jest unikatowy dla każdego użytkownika, ale w przeciwieństwie do soli, przechowywany w innymi miejscu niż skrót. Celem wprowadzenia tych argumentów jest utrudnienie intruzom przeprowadzenia ataków *łamania haseł* z zastosowaniem *tablic tęczowych*, czyli tablic zawierających wcześniej wyliczone (ang. *pre-computed*) skróty.

Metody oparte na wiedzy

Aby zapewnić odpowiedni poziom bezpieczeństwa skrótów, nawet dla *słabych* haseł, stosuje się ▸ rozciąganie kluczy (ang. *key stretching*), polegające na wielokrotnym, rekurencyjnym zastosowaniu algorytmu skracania hasła. Rekurencja w tym przypadku oznacza, że wynik algorytmu jest używany w następnym kroku jako jego argument. *Siła hasła* to jego atrybut, który decyduje jak trudno będzie poddawało się atakom mającym na celu jego złamanie. Porady dotyczące tworzenia haseł są np. publikowane przez ▸ CERT Polska.
Skróty są rozwiązaniem, które rozwiązuje problem bezpiecznego przechowywania haseł po stronie systemu uwierzytelniającego. Problemem pozostaje zabezpieczenie transmisji danych uwierzytelniających (ang. *credentials*), a także ich wprowadzania, które są narażone na ataki typu *man-in-the-middle*, *powtórzeniowe* (ang.*replay*), *phishing* i podsłuch (ang. *sniffing*) z użyciem złośliwego oprogramowania.

Metody oparte na wiedzy

Odpowiedzią na problem podsłuchu są *hasła częściowe* (ang. *partial passwords*) nazywane również w polskim środowisku informatycznym *hasłami maskowanymi*. Rozwiązanie to polega na tym, że użytkowniki podczas logowania podaje nie całe hasło, które ustalił w trakcie rejestracji lub które zostało mu nadane, ale tylko niektóre znaki znajdujące się na określonych pozycjach, np. 2, 10 i 16. Przy każdej próbie logowania użytkownik jest pytany o inne znaki z hasła. Po raz pierwszy ten ▸ mechanizm zastosowały banki, chcąc umożliwić klientom realizacje transakcji poprzez wykonanie telefonu i zlecenie ich pracownikom banku. To wymagało wcześniejszego uwierzytelnienia klienta, więc ustalano z nim wcześniej hasło. Ponieważ jednak cywilne przewodowe linie telefoniczne nie są zabezpieczone przed podsłuchem, podanie całego hasła mogło je ujawnić intruzowi. Dlatego pracownik nie pytał o całe hasło, ale tylko wybrane jego znaki.

Metody oparte na wiedzy

Podobne rozwiązanie zastosowano w systemach informatycznych. Niestety, nie jest to mechanizm prosty w implementacji. Nie tylko należy wcześniej przygotować odpowiednio dużą listę „masek” dla pojedynczego hasła i policzyć skróty dla każdej z nich z osobna, ale również odpowiednio dobrać długość takiego hasła i liczbę znaków, o którą system będzie pytał przy uwierzytelnianiu użytkownika. Całość hasła powinna być stosunkowo długa, np. 20 znaków. Natomiast liczba znaków, o które pyta system musi być na tyle krótka, aby atakujący mający możliwość podsłuchiwania łącza nie mógł ustalić hasła metodą zapisu (ang. *recording*) znaków przesyłanych, w kilku rundach logowania. Z drugiej strony ta liczba nie może być zbyt mała, aby atakujący nie mógł po prostu odgadnąć znaków (ang. *guessing*). Do tego dochodzi problem zależności między maskami, np. zbiór znaków z bieżącej maski może być podzbiorem znaków maski używanej wcześniej — liczba znaków do wprowadzenia nie musi być stała w każdym logowaniu.

Metody oparte na wiedzy

Skuteczniejszą ochroną przez przechwyceniem przesyłanego hasła jest zazwyczaj szyfrowanie łącza. Z kolei przez atakami typu *phishing*, czyli nakłonieniem użytkownika do użycia hasła w fałszywej wersji systemu, nad którym ma kontrolę intruz, chroni inny mechanizm, który wyświetla na stronie logowania element znany uprawnionemu użytkownikowi. Jeśli ten element się nie pojawia, to użytkownik powinien zrezygnować z dalszych czynności uwierzytelniania. Zazwyczaj takim elementem jest mały obraz, który użytkownik wybiera i zapamiętuje przy rejestracji w systemie. Następnie, kiedy loguje się do systemu, a dokładniej po identyfikacji, tj. wprowadzeniu loginu, na formatce wprowadzania hasła jest mu ten obrazek prezentowany. Aby mechanizm ten był skuteczny, liczba obrazków do wybrania powinna być odpowiednio dużą, rzędu dziesiątek.

Metody oparte na wiedzy

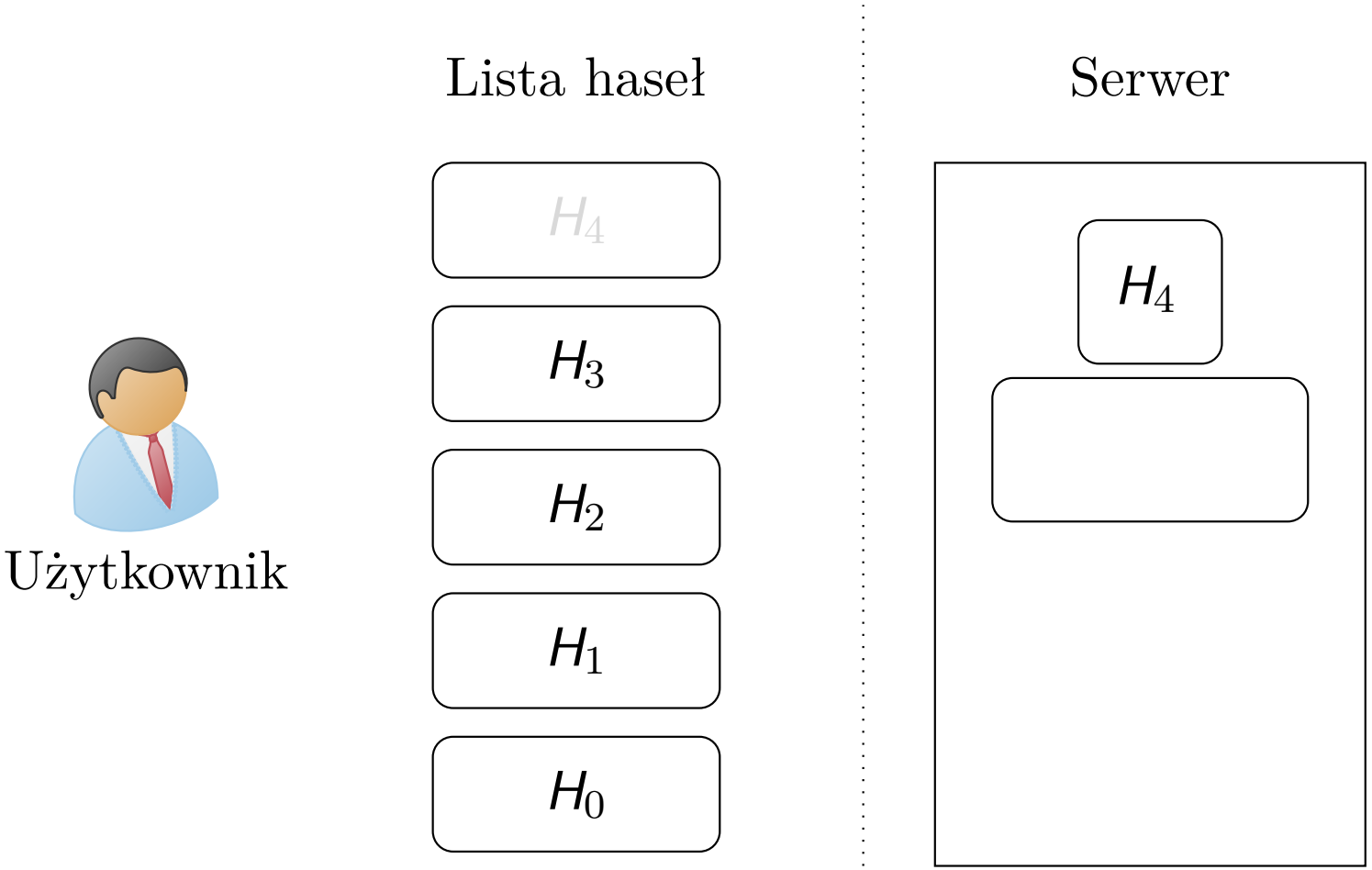
Głównym problemem związanym z metodami opartymi na wiedzy jest konieczność zapamiętania (w przypadku ludzi) lub przechowywania (w przypadku urządzeń) dużej liczby haseł/danych uwierzytelniających. W przypadku ludzi proponowanym rozwiązaniem, które może się sprawdzić są *menadżery haseł* (ang. *password managers*). Wątpliwości jakie budzą takie mechanizmy są związane z tym, że najczęściej są one budowane wbrew wypracowanym wcześniej zasadom bezpieczeństwa: hasła są przechowywane w formie zaszyfrowanej, a nie w formie kryptograficznego skrótu, jest jedno główne hasło, którego poznanie umożliwia dostęp do pozostałych. W końcu wiele menedżerów haseł zapisuje je w środowisku chmurowym, czyli w systemach znajdujących się poza kontrolą ich właściciela.

Metody oparte na posiadaniu

W przypadku metod opartych na posiadaniu weryfikacja tożsamości użytkownika polega na sprawdzeniu, czy ma on *token bezpieczeństwa*, który najczęściej jest wykonany w formie oprogramowania lub sprzętu. Udowodnienie posiadania tokena nie następuje bezpośrednio, ale poprzez zapytanie o wyniki operacji, którą ten token potrafi wykonać. Metody te opisywane są często jako bazujące na *czymś, co użytkownik ma*.

Mechanizmem pośrednim między metodami opartymi na wiedzy, a metodami opartymi na posiadaniu jest S/Key, którego działanie zostało dokładnie opisane w dokumencie ▸ RFC 1760 [2]. W tym rozwiązaniu użytkownik otrzymuje urządzenie lub listę haseł wygenerowanych przy pomocy funkcji skrótu. Każde z tych haseł może być użyte tylko jednokrotnie. Takie rozwiązanie chroni przed atakami powtórzeniowymi *replay*, ale nie przed atakami typu *man-in-the-middle*. Zasada jego działania została przedstawiona na Rysunku 1.

Metody oparte na posiadaniu



Schemat S/Key

Metody oparte na posiadaniu

Użytkownik

Lista haseł

H_4

H_3

H_2

H_1

H_0

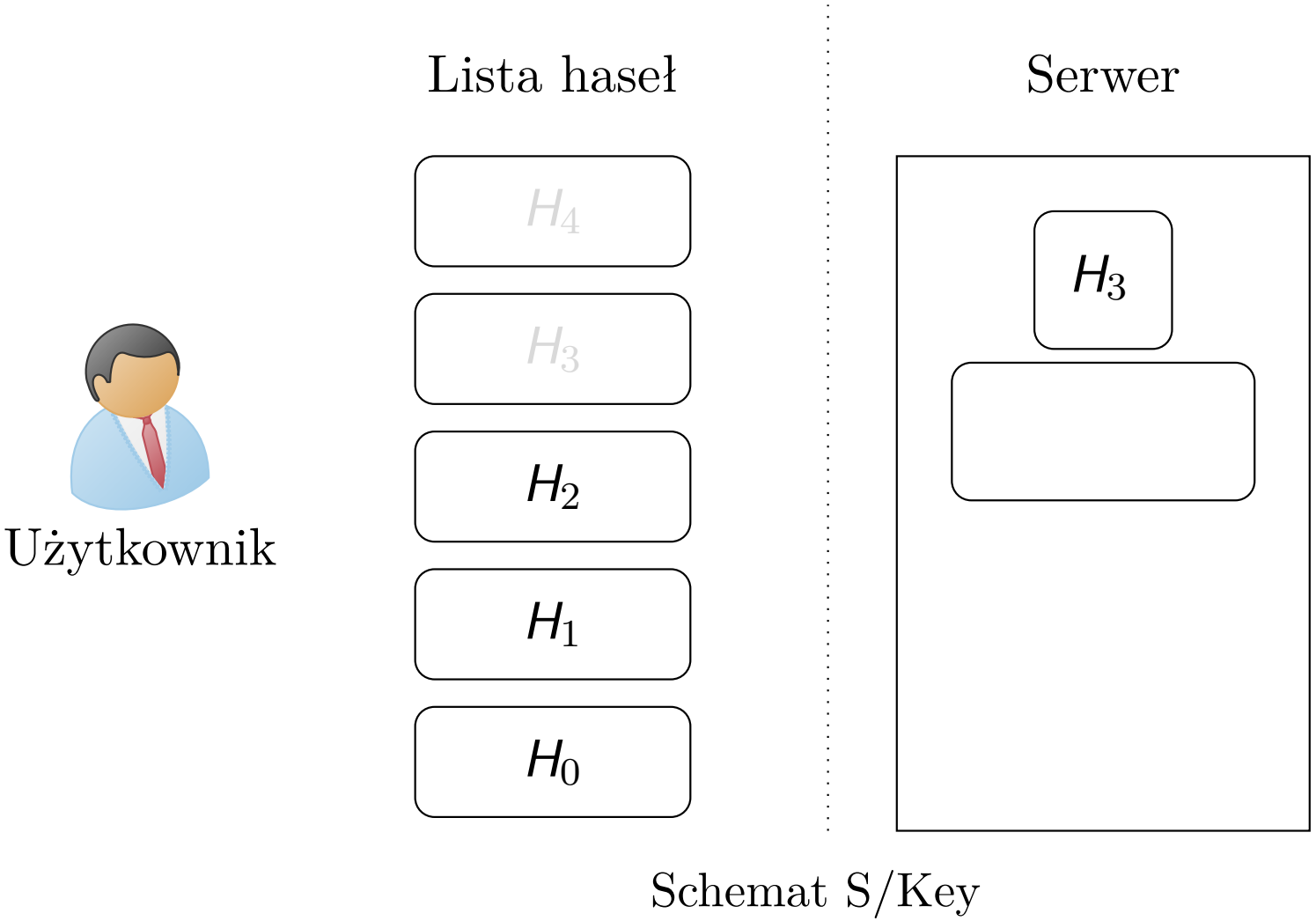
Serwer

H_4

$H(H_3) = H_4$

Schemat S/Key

Metody oparte na posiadaniu



Schemat S/Key

Metody oparte na posiadaniu

Użytkownik

Lista haseł

H_4

H_3

H_2

H_1

H_0

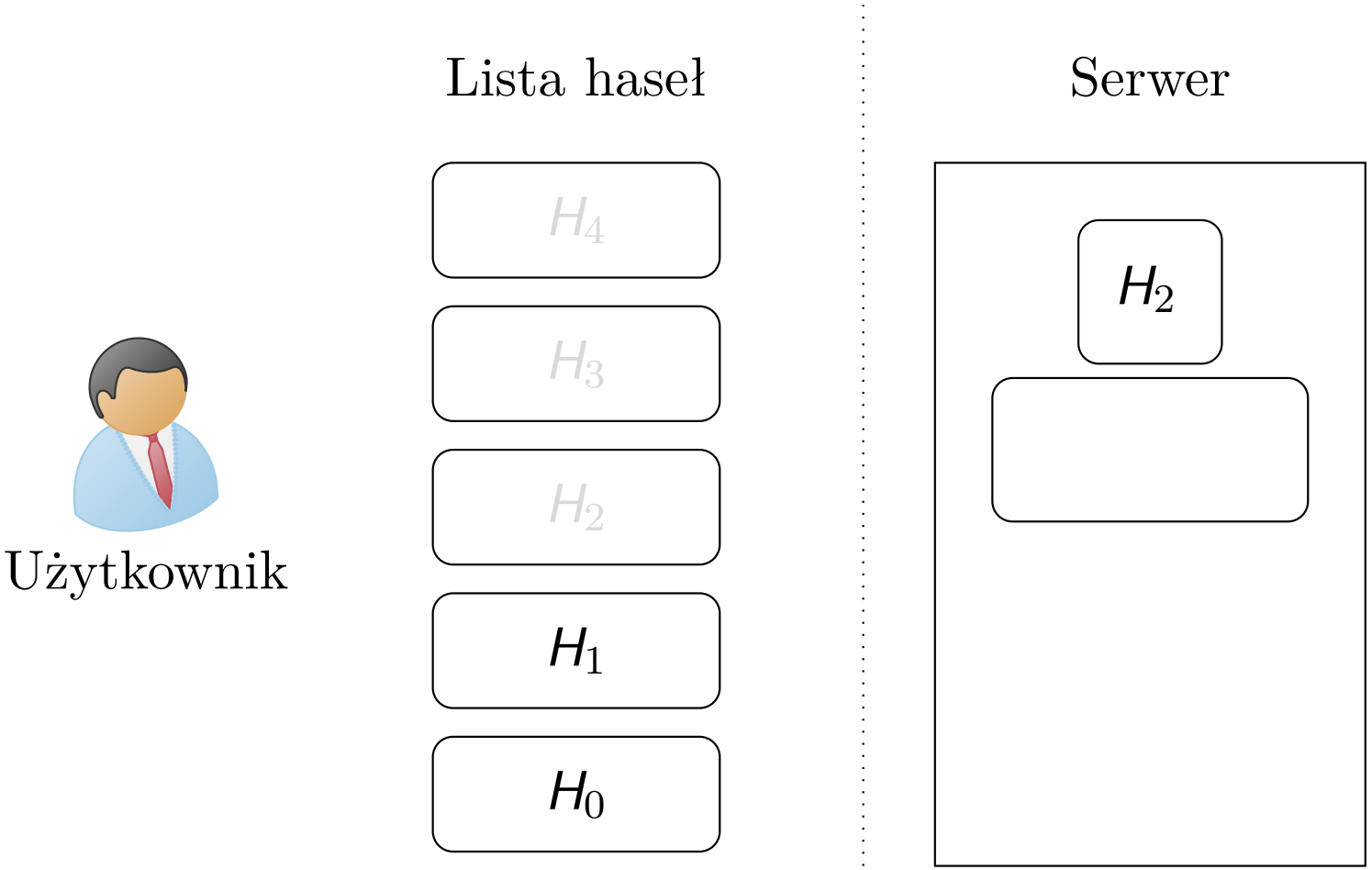
Serwer

H_3

$H(H_2) = H_3$

Schemat S/Key

Metody oparte na posiadaniu



Schemat S/Key

Metody oparte na posiadaniu

Użytkownik

Lista haseł

H_4

H_3

H_2

H_1

H_0

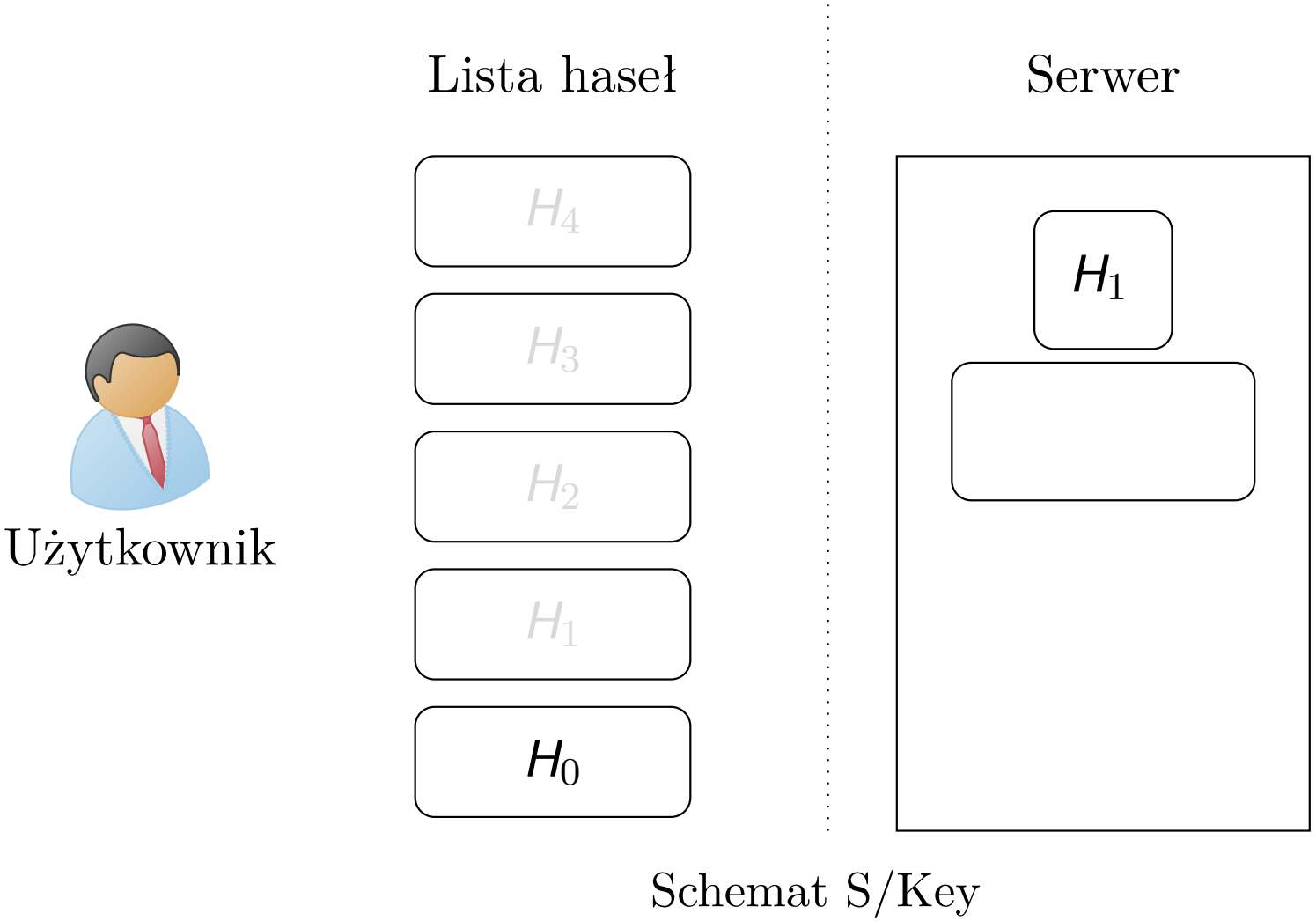
Serwer

H_2

$H(H_1) = H_2$

Schemat S/Key

Metody oparte na posiadaniu



Schemat S/Key

Metody oparte na posiadaniu

Użytkownik

Lista haseł

H_4

H_3

H_2

H_1

H_0

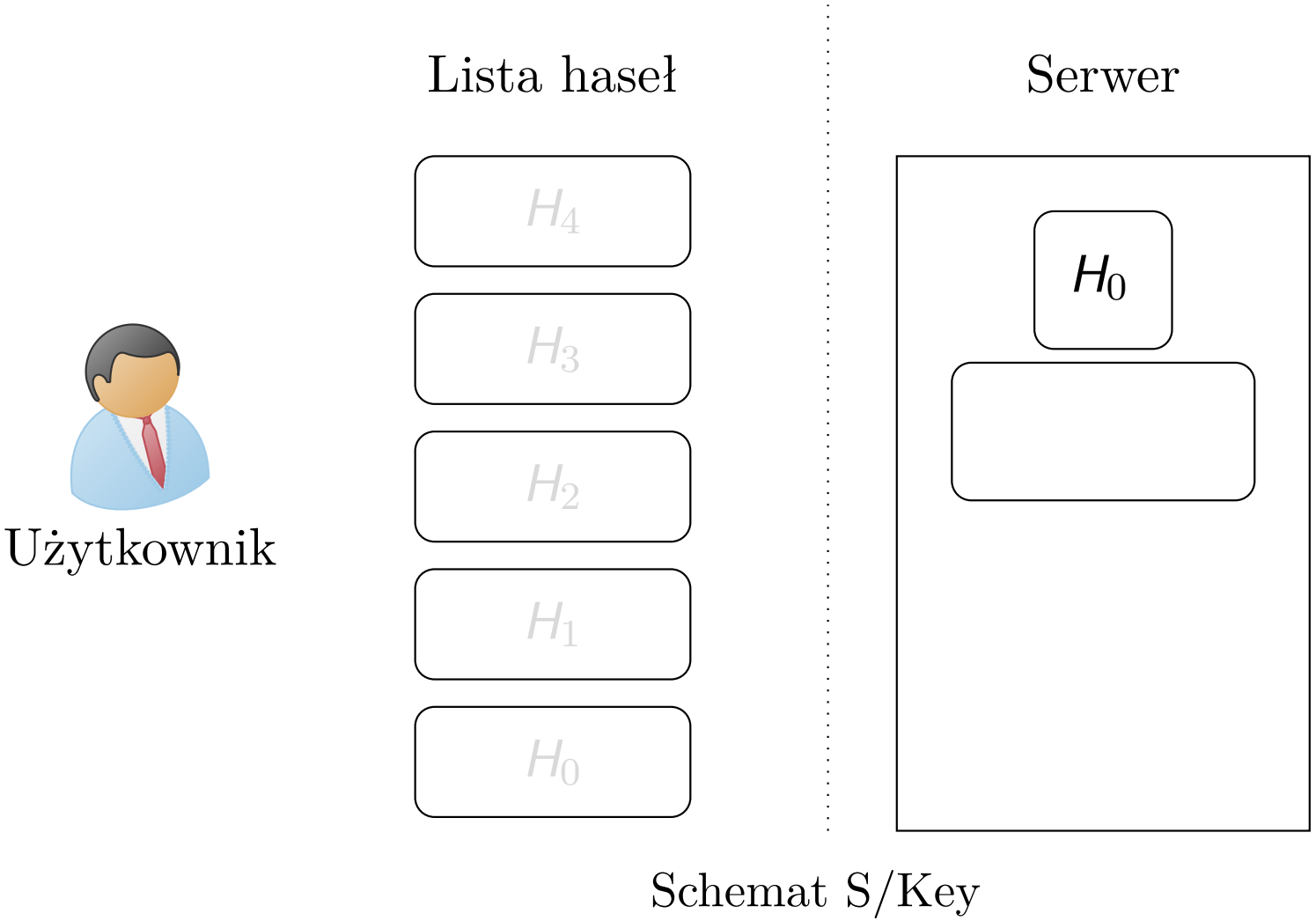
Serwer

H_1

$H(H_0) = H_1$

Schemat S/Key

Metody oparte na posiadaniu



Schemat S/Key

Metody oparte na posiadaniu

Proszę zwrócić uwagę, że po stronie serwera uwierzytelniającego zawsze przechowywane jest tylko jedno hasło, a użytkownik na swojej liście ma wszystkie, oprócz ostatniego. Jeśli lista ma faktyczną, fizyczną postać, a nie jest generowana na bieżąco przy pomocy tokena bezpieczeństwa, to pojawia się problem jej aktualizacji, gdy użytkownik wykorzysta wszystkie hasła.

Pomysł jednorazowych haseł (ang. *One-Time Password — OTP*) został opisany w dokumencie ▸ RFC2289. Mogą one być zrealizowane z użyciem protokołu typu *challenge-response*. Najprostsze rozwiązanie tego typu oparte jest na wiadomościach SMS. W serwerze uwierzytelniającym jest zarejestrowany numer telefonu użytkownika, którego tożsamość polega weryfikacji. Użytkownik celem jej potwierdzenia musi odesłać do serwera wartość, którą otrzymał od niego w SMSie. Niestety ze względu na słabości algorytmów szyfrowania używanych w telefonii komórkowej oraz na podatności protokołu SS7 oraz ataki typu SIM swap. Nie jest to bezpieczne rozwiązanie.

Metody oparte na posiadaniu

Inne realizacje koncepcji OTP polegają na wykorzystaniu protokołu typu *challenge*-response. W takim przypadku serwer uwierzytelniający i token mogą współdzielić sekretną wartość, która może być jednym z argumentów np. kryptograficznej funkcji skrótu. Celem uwierzytelnienia użytkownika serwer losuje unikatową wartość losową (ang. *nonce*), a następnie wysyła tę wartość do użytkownika. Ten ostatni oblicza jej skrót używając do tego tokena i wysyła go do serwera. Jeśli oba skróty, tj. ten obliczony przez użytkownika i serwer, są takie same, to tożsamość użytkownika jest potwierdzana.

Metody oparte na posiadaniu

W tym samym celu można wykorzystać kryptografię asymetryczną. Dodatkową korzyścią jest to, że serwer i token nie muszą współdzielić sekretnej wartości. Wystarczy, aby serwer znał klucz publiczny tokena, a token swój klucz prywatny. Protokół wykonywany jest podobnie jak w poprzednim rozwiązaniu. Serwer losuje unikatową wartość i szyfruje ją kluczem publicznym, a token odsyła jej postać jawną po zdeszyfrowaniu kluczem prywatnym. Do realizacji takiego mechanizmu można użyć kwalifikowanych certyfikatów, ale tylko tych, które wydawca specjalnie przeznaczył do tego celu.

Metody oparte na posiadaniu

Na szyfrowaniu asymetrycznym oparta jest także koncepcja *kluczy dostępu* ▸ (ang. *passkeys*) opracowana przez konsorcjum FIDO. W zasadzie sposób ich działania jest ▸ podobny do opisanego wcześniej, a różnice są tylko w szczegółach implementacji, takich jako konieczność dodatkowego uwierzytelnienia użytkownika (np. za pomocą biometrii) przez urządzenie przechowujące klucze prywatne, połączenie identyfikacji z uwierzytelnianiem oraz to, że klucze dostępu są realizowane na poziomie oprogramowania, a nie sprzętu.

Metody oparte na posiadaniu

Aby utrudnić przeprowadzanie ataków powtórzeniowych wprowadzono rozwiązanie typu *Time-Based One-Time Password* — TOPT, czyli takie, w których przydatność hasła jednorazowego jest ograniczona do zazwyczaj bardzo krótkiego czasu, rzędu minuty. Hasło, nazywane także *przepustką* (ang. *passcode*) jest sześciocyfrową liczbą losową, generowaną np. co minutę zarówno przez serwer uwierzytelniający, jak i token użytkownika przy pomocy kryptograficznie bezpiecznego generatora liczb pseudolosowych. Oba urządzenia mają wspólną wartość sekretną w postaci ziarna (ang. *seed*) będącego argumentem generatora. Ich zegary są za to niezależne. Uwierzytelnienie użytkownika polega na sprawdzeniu, czy jego token wygeneruje tę samą liczbę co serwer.

Metody oparte na posiadaniu

Problemem najczęściej okazuje się rozsynchronizowanie zegarów serwera i tokena. Dlatego najczęściej serwer sprawdzając wartość z tokena generuje nie tylko tę, której się spodziewa w danej minucie, ale również tę, która powinna wystąpić minutę później i minutę wcześniej. Jeśli któraś z tych ostatnich wykaże zgodność, to serwer odnotowuje w swojej bazie tokenów poprawkę dla tego urządzenia. Jeśli jednak rozbieżności są bardzo duże, to serwer może przeprowadzić procedurę ponownej synchronizacji z tokenem, która zazwyczaj polega na poproszeniu użytkownika o podanie kilku kolejnych wartości generowanych przez to urządzenie. Trwałość takich tokenów również może być kłopotliwa. Rysunek 2 przedstawia token sprzętowy SecurID (mogą też mieć postać aplikacji), generujący TOTP.

Metody oparte na posiadaniu

Token SecurID (źródło: https://en.wikipedia.org/wiki/RSA_SecurID)

Metody oparte na cechach/atrybutach

Metody oparte na cechach/atrybutach można opisać jako bazujące na koncepcji *to, kim/czym użytkownik jest*. W przypadku uwierzytelniania ludzi sprowadzają się one do zastosowania *biometrii*, która może polegać na badaniu geometrii dłoni lub twarzy, odcisków palców, budowy układu krwionośnego, analizie głosu lub skanowaniu siatkówki oka. Pozornie wydają się to bezpieczne i wygodne metody, ale bazują one głównie na statystyce, więc nie są pewne. Znane są przypadki, gdy np. analizator głosu mylił dwie osoby lub przestawał rozpoznawać chorych użytkowników. Najbardziej niezawodna z tych metod, jest skanowanie siatkówki oka, ale nawet ona nie jest odporna na próby złamania z użyciem nowoczesnych kamer lub aparatów fotograficznych. Podobne problemy dotyczą także ▸ rozpoznawania twarzy. Dodatkową wadą tych metod jest niemalże całkowity brak zmiany sposobu weryfikacji tożsamości oraz możliwość wykluczenia osób niepełnosprawnych.

Metody oparte na cechach/atrybutach

Niektóre metody biometryczne pozwalają na uwierzytelnianie użytkownika nie tylko jednorazowo, w trakcie rozpoczynania przez niego pracy z systemem, ale również w sposób ciągły. Do takich metod zalicza się np. badanie statystyki pisania na klawiaturze. Niemniej „typowe” metody związane z biometrią powinny być stosowane razem z innymi metodami uwierzytelniania.

Metody oparte na lokalizacji

Metody te bazują na badaniu gdzie w chwili uwierzytelniania znajduje się użytkownik. Ta informacja może mieć postać współrzędnych GPS, adresu IP lub domeny sieci i zazwyczaj jest też porównywana z czasem logowania. Jeśli użytkownik loguje się np. z Warszawy, a chwilę później z Rio De Janeiro, to jest to wyraźny sygnał, że może chodzić o próbę włamania do systemu. Te metody nie mogą być stosowane samodzielnie, muszą być używane w połączeniu z innymi mechanizmami weryfikowania tożsamości.

Metody oparte na zachowaniu

Metody operate na zachowaniu badają *w sposób ciągły* to co użytkownik robi. Dzięki temu na bieżąco monitorują jego położenie oraz mogą rozpoznać kiedy chce korzystać z usług chronionego systemu. Są one subkategorią metod biometrycznych i mogą polegać na badaniu sposobu korzystania przez użytkownika z klawiatury i myszki, sposobie jego chodu, tego jak tworzy zdania, jak posługuje się językiem naturalnym itp. Dane dla tych metod mogą być zbierane przy pomocy telefonów komórkowych oraz innych urządzeń zaliczanych do grupy *wearable*. Rozwiązania te są kontrowersyjne, bo mogą naruszać prywatność użytkowników.

Podsumowanie

Część z opisanych metod w zasadzie nie nadaje się do samodzielnego zastosowania. Inne zyskują na niezawodności, jeśli są używane w połączeniu z pozostałymi. Dlatego obecnie zaleca się stosowanie uwierzytelniania wieloskładnikowego (MFA).

Dodatkowe źródła

- Michał Sajdak i inni. *Wprowadzenie do bezpieczeństwa IT*. Tom 2. Securitum, Kraków, 2024.
- Tomasz Surmacz. *Secure Systems and Networks*. PRINTPAP, Łódź, 2011. URL: https://www.dbc.wroc.pl/Content/23915/PDF/Surmacz_Secure_Systems.pdf.

Pytania

?

Koniec

Dziękuję Państwu za uwagę!