Kryptografia

z elementami kryptografii kwantowej

Ryszard Tanaś

http://zon8.physd.amu.edu.pl/~tanas

Wykład 10

Spis treści

15 Uwierzytelnianie

- Kluczową sprawą dla bezpieczeństwa systemów komputerowych jest zapewnienie dostępu do systemu i zasobów tylko osobom do tego uprawnionym.
- W systemie musi więc być wbudowany mechanizm sprawdzania, czy użytkownik podający się za Alicję, naprawdę nią jest.
- Do tego celu służy mechanizm uwierzytelniania lub identyfikacji (tutaj nie rozróżniamy tych pojęć, chociaż czasem się je rozróżnia).

15 Uwierzytelnianie

- Kluczową sprawą dla bezpieczeństwa systemów komputerowych jest zapewnienie dostępu do systemu i zasobów tylko osobom do tego uprawnionym.
- W systemie musi więc być wbudowany mechanizm sprawdzania, czy użytkownik podający się za Alicję, naprawdę nią jest.
- Do tego celu służy mechanizm uwierzytelniania lub identyfikacji (tutaj nie rozróżniamy tych pojęć, chociaż czasem się je rozróżnia).

15 Uwierzytelnianie

- Kluczową sprawą dla bezpieczeństwa systemów komputerowych jest zapewnienie dostępu do systemu i zasobów tylko osobom do tego uprawnionym.
- W systemie musi więc być wbudowany mechanizm sprawdzania, czy użytkownik podający się za Alicję, naprawdę nią jest.
- Do tego celu służy mechanizm uwierzytelniania lub identyfikacji (tutaj nie rozróżniamy tych pojęć, chociaż czasem się je rozróżnia).

15 Uwierzytelnianie

- Kluczową sprawą dla bezpieczeństwa systemów komputerowych jest zapewnienie dostępu do systemu i zasobów tylko osobom do tego uprawnionym.
- W systemie musi więc być wbudowany mechanizm sprawdzania, czy użytkownik podający się za Alicję, naprawdę nią jest.
- Do tego celu służy mechanizm uwierzytelniania lub identyfikacji (tutaj nie rozróżniamy tych pojęć, chociaż czasem się je rozróżnia).

15.1 Hasła

Najczęściej stosowany system identyfikacji to system haseł. Alicja chcąc wejść do sytemu podaje tajne hasło znane tylko jej i systemowi.

15.1.1 Hasła w systemie Unix

- Hasła w systemie Unix szyfrowane są programem `crypt`, który stanowi pewną modyfikację DES.
- Użytkownik wybiera ośmioliterowe hasło.
- Z każdego bajtu reprezentującego literę hasła wybieranych jest 7 bitów, które w rezultacie tworzą 56 bitowy klucz.

15.1 Hasła

Najczęściej stosowany system identyfikacji to system haseł. Alicja chcąc wejść do sytemu podaje tajne hasło znane tylko jej i systemowi.

15.1.1 Hasła w systemie Unix

- Hasła w systemie Unix szyfrowane są programem `crypt`, który stanowi pewną modyfikację DES.
- Użytkownik wybiera ośmioliterowe hasło.
- Z każdego bajtu reprezentującego literę hasła wybieranych jest 7 bitów, które w rezultacie tworzą 56 bitowy klucz.

15.1 Hasła

Najczęściej stosowany system identyfikacji to system haseł. Alicja chcąc wejść do sytemu podaje tajne hasło znane tylko jej i systemowi.

15.1.1 Hasła w systemie Unix

- Hasła w systemie Unix szyfrowane są programem `crypt`, który stanowi pewną modyfikację DES.
- Użytkownik wybiera ośmioliterowe hasło.
- Z każdego bajtu reprezentującego literę hasła wybieranych jest 7 bitów, które w rezultacie tworzą 56 bitowy klucz.

15.1 Hasła

Najczęściej stosowany system identyfikacji to system haseł. Alicja chcąc wejść do sytemu podaje tajne hasło znane tylko jej i systemowi.

15.1.1 Hasła w systemie Unix

- Hasła w systemie Unix szyfrowane są programem `crypt`, który stanowi pewną modyfikację DES.
- Użytkownik wybiera ośmioliterowe hasło.
- Z każdego bajtu reprezentującego literę hasła wybieranych jest 7 bitów, które w rezultacie tworzą 56 bitowy klucz.

- Klucz ten służy do szyfrowania 64 bitowego bloku znanego tekstu (zwykle same zera). Wynik podlega kolejnemu szyfrowaniu, i tak 25 razy.
- Dodatkowo używa się 12 bitów („salt") generowanych przez zegar systemowy w momencie tworzenia hasła. Bity te są wykorzystane w permutacji rozszerzającej DES.
- Wynik szyfrowania (64 bity) plus „salt" (12 bitów) jest „przepakowany" i zapisywany w postaci 11 znaków ASCII.
- Hasło przechowywane jest w postaci 13 znaków ASCII, które zawierają dwa znaki „salt" oraz 11 znaków zaszyfrowanego hasła.

- Klucz ten służy do szyfrowania 64 bitowego bloku znanego tekstu (zwykle same zera). Wynik podlega kolejnemu szyfrowaniu, i tak 25 razy.
- Dodatkowo używa się 12 bitów („salt") generowanych przez zegar systemowy w momencie tworzenia hasła. Bity te są wykorzystane w permutacji rozszerzającej DES.
- Wynik szyfrowania (64 bity) plus „salt" (12 bitów) jest „przepakowany" i zapisywany w postaci 11 znaków ASCII.
- Hasło przechowywane jest w postaci 13 znaków ASCII, które zawierają dwa znaki „salt" oraz 11 znaków zaszyfrowanego hasła.

- Klucz ten służy do szyfrowania 64 bitowego bloku znanego tekstu (zwykle same zera). Wynik podlega kolejnemu szyfrowaniu, i tak 25 razy.
- Dodatkowo używa się 12 bitów („salt") generowanych przez zegar systemowy w momencie tworzenia hasła. Bity te są wykorzystane w permutacji rozszerzającej DES.
- Wynik szyfrowania (64 bity) plus „salt" (12 bitów) jest „przepakowany" i zapisywany w postaci 11 znaków ASCII.
- Hasło przechowywane jest w postaci 13 znaków ASCII, które zawierają dwa znaki „salt" oraz 11 znaków zaszyfrowanego hasła.

- Klucz ten służy do szyfrowania 64 bitowego bloku znanego tekstu (zwykle same zera). Wynik podlega kolejnemu szyfrowaniu, i tak 25 razy.
- Dodatkowo używa się 12 bitów („salt") generowanych przez zegar systemowy w momencie tworzenia hasła. Bity te są wykorzystane w permutacji rozszerzającej DES.
- Wynik szyfrowania (64 bity) plus „salt" (12 bitów) jest „przepakowany" i zapisywany w postaci 11 znaków ASCII.
- Hasło przechowywane jest w postaci 13 znaków ASCII, które zawierają dwa znaki „salt" oraz 11 znaków zaszyfrowanego hasła.

- Dodanie 12 bitów „salt” powoduje, że liczba możliwości dla wybranego hasła zwiększa się $2^{12} = 4096$ razy.
- W nowszych systemach stosuje się bezpieczniejsze sposoby szyfrowania haseł, np. algorytm MD5

- Dodanie 12 bitów „salt” powoduje, że liczba możliwości dla wybranego hasła zwiększa się $2^{12} = 4096$ razy.
- W nowszych systemach stosuje się bezpieczniejsze sposoby szyfrowania haseł, np. algorytm MD5

15.2 PIN — Personal Identification Number

- Odmianą hasła jest także PIN używany w przypadku kart kredytowych, bankowych, czy tzw. tokenów.
- Jest to zwykle liczba czterocyfrowa (czasem ośmiocyfrowa), która ma zabezpieczać przed użyciem karty przez osoby niepowołane, np. złodzieja.

15.2 PIN — Personal Identification Number

- Odmianą hasła jest także PIN używany w przypadku kart kredytowych, bankowych, czy tzw. tokenów.
- Jest to zwykle liczba czterocyfrowa (czasem ośmiocyfrowa), która ma zabezpieczać przed użyciem karty przez osoby niepowołane, np. złodzieja.

15.2 PIN — Personal Identification Number

- Odmianą hasła jest także PIN używany w przypadku kart kredytowych, bankowych, czy tzw. tokenów.
- Jest to zwykle liczba czterocyfrowa (czasem ośmiocyfrowa), która ma zabezpieczać przed użyciem karty przez osoby niepowołane, np. złodzieja.

15.3 Protokół challenge-response

- Idea tego sposobu identyfikacji polega na odpowiedzi (response) Alicji na wyzwanie (challenge) przesłane przez Bolka, która przekona Bolka, że ma do czynienie rzeczywiście z Alicją.

15.3 Protokół challenge-response

- Idea tego sposobu identyfikacji polega na odpowiedzi (response) Alicji na wyzwanie (challenge) przesłane przez Bolka, która przekona Bolka, że ma do czynienie rzeczywiście z Alicją.

15.3.1 Protokół challenge-response z tajnym kluczem

- Alicja i Bolek dysponują takim samym tajnym kluczem K (algorytm symetryczny) oraz umówili się jakiej funkcji hashującej H będą używać.
- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja oblicza $H(K, r_B)$ i przesyła wynik Bolkowi
- Bolek także oblicza $H(K, r_B)$ i jeśli wynik zgadza się z wynikiem przysłanym przez Alicję to tożsamość Alicji zostaje potwierdzona

15.3.1 Protokół challenge-response z tajnym kluczem

- Alicja i Bolek dysponują takim samym tajnym kluczem K (algorytm symetryczny) oraz umówili się jakiej funkcji hashującej H będą używać.
- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja oblicza $H(K, r_B)$ i przesyła wynik Bolkowi
- Bolek także oblicza $H(K, r_B)$ i jeśli wynik zgadza się z wynikiem przysłanym przez Alicję to tożsamość Alicji zostaje potwierdzona

15.3.1 Protokół challenge-response z tajnym kluczem

- Alicja i Bolek dysponują takim samym tajnym kluczem K (algorytm symetryczny) oraz umówili się jakiej funkcji hashującej H będą używać.
- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja oblicza $H(K, r_B)$ i przesyła wynik Bolkowi
- Bolek także oblicza $H(K, r_B)$ i jeśli wynik zgadza się z wynikiem przysłanym przez Alicję to tożsamość Alicji zostaje potwierdzona

15.3.1 Protokół challenge-response z tajnym kluczem

- Alicja i Bolek dysponują takim samym tajnym kluczem K (algorytm symetryczny) oraz umówili się jakiej funkcji hashującej H będą używać.
- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja oblicza $H(K, r_B)$ i przesyła wynik Bolkowi
- Bolek także oblicza $H(K, r_B)$ i jeśli wynik zgadza się z wynikiem przysłanym przez Alicję to tożsamość Alicji zostaje potwierdzona

15.3.1 Protokół challenge-response z tajnym kluczem

- Alicja i Bolek dysponują takim samym tajnym kluczem K (algorytm symetryczny) oraz umówili się jakiej funkcji hashującej H będą używać.
- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja oblicza $H(K, r_B)$ i przesyła wynik Bolkowi
- Bolek także oblicza $H(K, r_B)$ i jeśli wynik zgadza się z wynikiem przysłanym przez Alicję to tożsamość Alicji zostaje potwierdzona

15.3.1 Protokół challenge-response z tajnym kluczem

- Alicja i Bolek dysponują takim samym tajnym kluczem K (algorytm symetryczny) oraz umówili się jakiej funkcji hashującej H będą używać.
- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja oblicza $H(K, r_B)$ i przesyła wynik Bolkowi
- Bolek także oblicza $H(K, r_B)$ i jeśli wynik zgadza się z wynikiem przysłanym przez Alicję to tożsamość Alicji zostaje potwierdzona

15.3.2 Protokół challenge-response z kluczem publicznym

- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja szyfruje liczbę r_B używając swojego klucza prywatnego i kryptogram wysyła do Bolka
- Bolek deszyfruje kryptogram otrzymany od Alicji używając jej klucza publicznego i jeśli w wyniku otrzyma r_B to tożsamość Alicji jest potwierdzona

15.3.2 Protokół challenge-response z kluczem publicznym

- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja szyfruje liczbę r_B używając swojego klucza prywatnego i kryptogram wysyła do Bolka
- Bolek deszyfruje kryptogram otrzymany od Alicji używając jej klucza publicznego i jeśli w wyniku otrzyma r_B to tożsamość Alicji jest potwierdzona

15.3.2 Protokół challenge-response z kluczem publicznym

- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja szyfruje liczbę r_B używając swojego klucza prywatnego i kryptogram wysyła do Bolka
- Bolek deszyfruje kryptogram otrzymany od Alicji używając jej klucza publicznego i jeśli w wyniku otrzyma r_B to tożsamość Alicji jest potwierdzona

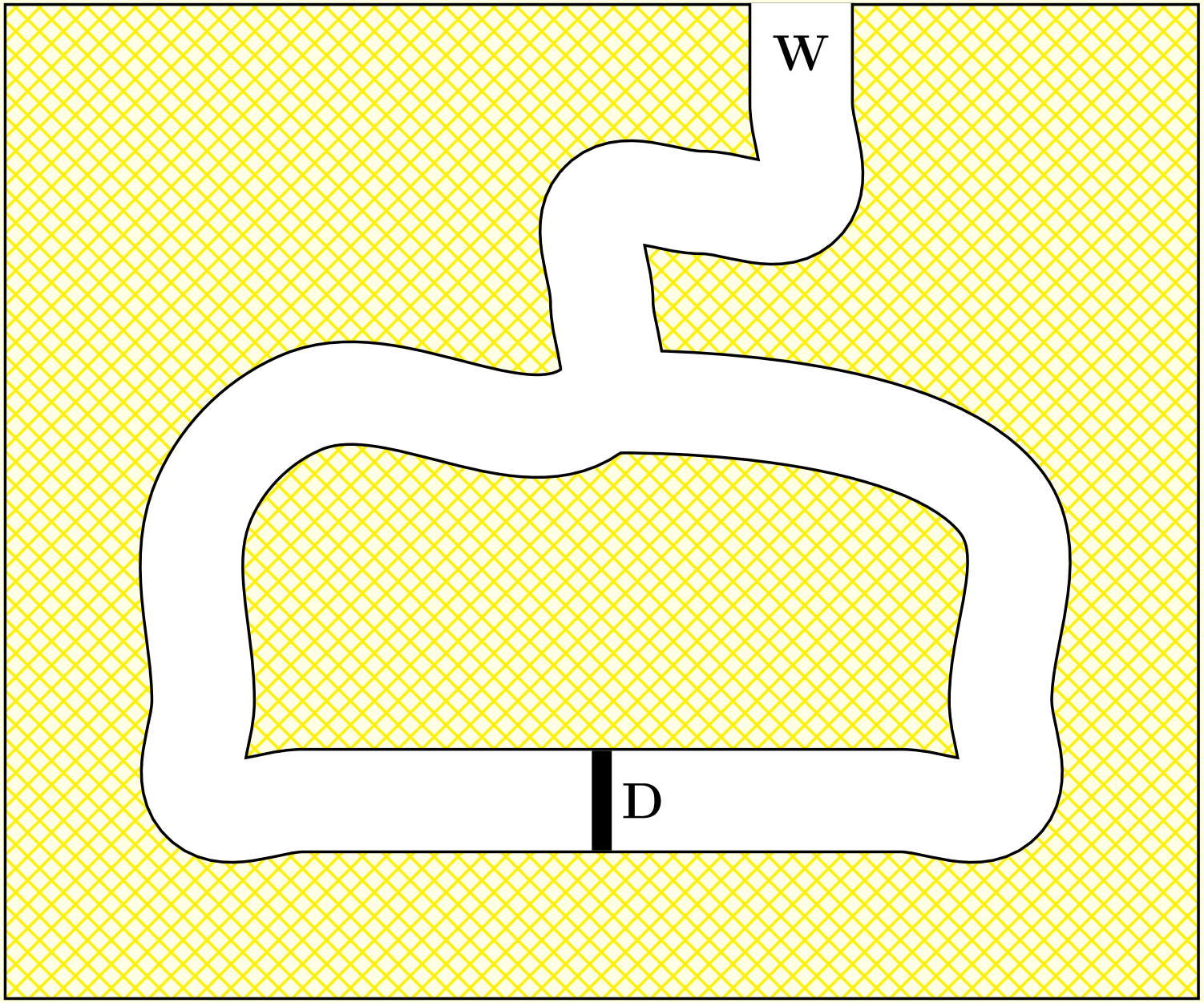
15.3.2 Protokół challenge-response z kluczem publicznym

- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja szyfruje liczbę r_B używając swojego klucza prywatnego i kryptogram wysyła do Bolka
- Bolek deszyfruje kryptogram otrzymany od Alicji używając jej klucza publicznego i jeśli w wyniku otrzyma r_B to tożsamość Alicji jest potwierdzona

15.3.2 Protokół challenge-response z kluczem publicznym

- Alicja komunikuje się z Bolkiem przedstawiając się jako Alicja
- Bolek generuje liczbę losową r_B i wysyła ją Alicji
- Alicja szyfruje liczbę r_B używając swojego klucza prywatnego i kryptogram wysyła do Bolka
- Bolek deszyfruje kryptogram otrzymany od Alicji używając jej klucza publicznego i jeśli w wyniku otrzyma r_B to tożsamość Alicji jest potwierdzona

15.4 Dowody z wiedzą zerową



Jaskinia

- Alicja chce przekonać Bolka, że zna pewien sekret, ale nie chce zdradzić samego sekretu.
  Alicja twierdzi, że potrafi otworzyć drzwi zamykające przejście w jaskini.
- Bolek stoi przy wejściu do jaskini
- Alicja wchodzi do jaskini i idzie albo w lewo albo w prawo dochodząc do drzwi zamykających przejście
- Bolek dochodzi do rozwidlenia korytarza, rzuca monetą i w zależności od wyniku rzutu krzyczy, nakazując Alicji wyjść albo z lewego korytarza albo z prawego
- Alicja wykonuje polecenie Bolka, otwierając drzwi jeśli to konieczne

- Alicja chce przekonać Bolka, że zna pewien sekret, ale nie chce zdradzić samego sekretu.
  Alicja twierdzi, że potrafi otworzyć drzwi zamykające przejście w jaskini.
- Bolek stoi przy wejściu do jaskini
- Alicja wchodzi do jaskini i idzie albo w lewo albo w prawo dochodząc do drzwi zamykających przejście
- Bolek dochodzi do rozwidlenia korytarza, rzuca monetą i w zależności od wyniku rzutu krzyczy, nakazując Alicji wyjść albo z lewego korytarza albo z prawego
- Alicja wykonuje polecenie Bolka, otwierając drzwi jeśli to konieczne

- Alicja chce przekonać Bolka, że zna pewien sekret, ale nie chce zdradzić samego sekretu.
  Alicja twierdzi, że potrafi otworzyć drzwi zamykające przejście w jaskini.
- Bolek stoi przy wejściu do jaskini
- Alicja wchodzi do jaskini i idzie albo w lewo albo w prawo dochodząc do drzwi zamykających przejście
- Bolek dochodzi do rozwidlenia korytarza, rzuca monetą i w zależności od wyniku rzutu krzyczy, nakazując Alicji wyjść albo z lewego korytarza albo z prawego
- Alicja wykonuje polecenie Bolka, otwierając drzwi jeśli to konieczne

- Alicja chce przekonać Bolka, że zna pewien sekret, ale nie chce zdradzić samego sekretu.
  Alicja twierdzi, że potrafi otworzyć drzwi zamykające przejście w jaskini.
- Bolek stoi przy wejściu do jaskini
- Alicja wchodzi do jaskini i idzie albo w lewo albo w prawo dochodząc do drzwi zamykających przejście
- Bolek dochodzi do rozwidlenia korytarza, rzuca monetą i w zależności od wyniku rzutu krzyczy, nakazując Alicji wyjść albo z lewego korytarza albo z prawego
- Alicja wykonuje polecenie Bolka, otwierając drzwi jeśli to konieczne

- Alicja chce przekonać Bolka, że zna pewien sekret, ale nie chce zdradzić samego sekretu.
  Alicja twierdzi, że potrafi otworzyć drzwi zamykające przejście w jaskini.
- Bolek stoi przy wejściu do jaskini
- Alicja wchodzi do jaskini i idzie albo w lewo albo w prawo dochodząc do drzwi zamykających przejście
- Bolek dochodzi do rozwidlenia korytarza, rzuca monetą i w zależności od wyniku rzutu krzyczy, nakazując Alicji wyjść albo z lewego korytarza albo z prawego
- Alicja wykonuje polecenie Bolka, otwierając drzwi jeśli to konieczne

- Doświadczenie takie powtarzają n-krotnie. Jeśli n jest dostatecznie duże, to prawdopodobieństwo tego, że Alicja wykona polecenie Bolka nie potrafiąc otworzyć drzwi jest znikomo małe ($1/2^n$).

15.5 Dowód o wiedzy zerowej dla logarytmu dyskretnego

- Alicja chce przekonać Bolka, że zna wartość logarytmu dyskretnego bez zdradzanie tej wartości. Czyli chce udowodnić, że zna liczbę x, która spełnia zależność $a^x = b \pmod p$, gdzie p jest dużą liczbą pierwszą. Oboje znają $\{p, a, b\}$, natomiast Bolek nie zna x.
- Alicja generuje t liczb losowych $r_1, r_2, \ldots, r_t$ mniejszych od $p - 1$
- Alicja oblicza $h_i \equiv a^{r_i} \pmod p$ i przesyła je Bolkowi
- Alicja i Bolek wspólnie rzucają t razy monetą generując w ten sposób t bitów $b_1, b_2, \ldots, b_t$

15.5 Dowód o wiedzy zerowej dla logarytmu dyskretnego

- Alicja chce przekonać Bolka, że zna wartość logarytmu dyskretnego bez zdradzanie tej wartości. Czyli chce udowodnić, że zna liczbę x, która spełnia zależność $a^x = b \pmod p$, gdzie p jest dużą liczbą pierwszą. Oboje znają $\{p, a, b\}$, natomiast Bolek nie zna x.
- Alicja generuje t liczb losowych $r_1, r_2, \ldots, r_t$ mniejszych od $p - 1$
- Alicja oblicza $h_i \equiv a^{r_i} \pmod p$ i przesyła je Bolkowi
- Alicja i Bolek wspólnie rzucają t razy monetą generując w ten sposób t bitów $b_1, b_2, \ldots, b_t$

15.5 Dowód o wiedzy zerowej dla logarytmu dyskretnego

- Alicja chce przekonać Bolka, że zna wartość logarytmu dyskretnego bez zdradzanie tej wartości. Czyli chce udowodnić, że zna liczbę x, która spełnia zależność $a^x = b \pmod p$, gdzie p jest dużą liczbą pierwszą. Oboje znają $\{p, a, b\}$, natomiast Bolek nie zna x.
- Alicja generuje t liczb losowych $r_1, r_2, \ldots, r_t$ mniejszych od $p - 1$
- Alicja oblicza $h_i \equiv a^{r_i} \pmod p$ i przesyła je Bolkowi
- Alicja i Bolek wspólnie rzucają t razy monetą generując w ten sposób t bitów $b_1, b_2, \ldots, b_t$

15.5 Dowód o wiedzy zerowej dla logarytmu dyskretnego

- Alicja chce przekonać Bolka, że zna wartość logarytmu dyskretnego bez zdradzanie tej wartości. Czyli chce udowodnić, że zna liczbę x, która spełnia zależność $a^x = b \pmod p$, gdzie p jest dużą liczbą pierwszą. Oboje znają $\{p, a, b\}$, natomiast Bolek nie zna x.
- Alicja generuje t liczb losowych $r_1, r_2, \ldots, r_t$ mniejszych od $p - 1$
- Alicja oblicza $h_i \equiv a^{r_i} \pmod p$ i przesyła je Bolkowi
- Alicja i Bolek wspólnie rzucają t razy monetą generując w ten sposób t bitów $b_1, b_2, \ldots, b_t$

15.5 Dowód o wiedzy zerowej dla logarytmu dyskretnego

- Alicja chce przekonać Bolka, że zna wartość logarytmu dyskretnego bez zdradzanie tej wartości. Czyli chce udowodnić, że zna liczbę x, która spełnia zależność $a^x = b \pmod p$, gdzie p jest dużą liczbą pierwszą. Oboje znają $\{p, a, b\}$, natomiast Bolek nie zna x.
- Alicja generuje t liczb losowych $r_1, r_2, \ldots, r_t$ mniejszych od $p - 1$
- Alicja oblicza $h_i \equiv a^{r_i} \pmod p$ i przesyła je Bolkowi
- Alicja i Bolek wspólnie rzucają t razy monetą generując w ten sposób t bitów $b_1, b_2, \ldots, b_t$

- Dla wszystkich bitów Alicja oblicza i przesyła Bolkowi następujące liczby

  r_i jeśli $b_i = 0$

  $s_i = r_i - r_j$ jeśli $b_i = 1$

  gdzie j jest największą wartością, dla której $b_j = 1$

- Dla wszystkich bitów t Bolek sprawdza czy

  $a^{r_i} \equiv h_i \pmod{p}$ dla $b_i = 0$

  $a^{s_i} \equiv h_i h_j^{-1} \pmod{p}$ dla $b_i = 1$

- Dla każdego i, dla którego $b_i = 1$, Alicja oblicza i wysyła Bolkowi $z_i = (x - r_i) \pmod{p-1}$

- Bolek sprawdza czy $a^{z_i} \equiv b h_i^{-1} \pmod{p}$

- Dla wszystkich bitów Alicja oblicza i przesyła Bolkowi następujące liczby

  r_i jeśli $b_i = 0$

  $s_i = r_i - r_j$ jeśli $b_i = 1$

  gdzie j jest największą wartością, dla której $b_j = 1$

- Dla wszystkich bitów t Bolek sprawdza czy

  $a^{r_i} \equiv h_i \pmod{p}$ dla $b_i = 0$

  $a^{s_i} \equiv h_i h_j^{-1} \pmod{p}$ dla $b_i = 1$

- Dla każdego i, dla którego $b_i = 1$, Alicja oblicza i wysyła Bolkowi $z_i = (x - r_i) \pmod{p-1}$

- Bolek sprawdza czy $a^{z_i} \equiv b h_i^{-1} \pmod{p}$

- Dla wszystkich bitów Alicja oblicza i przesyła Bolkowi następujące liczby

  r_i jeśli $b_i = 0$

  $s_i = r_i - r_j$ jeśli $b_i = 1$

  gdzie j jest największą wartością, dla której $b_j = 1$

- Dla wszystkich bitów t Bolek sprawdza czy

  $a^{r_i} \equiv h_i \pmod{p}$ dla $b_i = 0$

  $a^{s_i} \equiv h_i h_j^{-1} \pmod{p}$ dla $b_i = 1$

- Dla każdego i, dla którego $b_i = 1$, Alicja oblicza i wysyła Bolkowi $z_i = (x - r_i) \pmod{p-1}$

- Bolek sprawdza czy $a^{z_i} \equiv b h_i^{-1} \pmod{p}$

- Dla wszystkich bitów Alicja oblicza i przesyła Bolkowi następujące liczby

  r_i jeśli $b_i = 0$

  $s_i = r_i - r_j$ jeśli $b_i = 1$

  gdzie j jest największą wartością, dla której $b_j = 1$

- Dla wszystkich bitów t Bolek sprawdza czy

  $a^{r_i} \equiv h_i \pmod{p}$ dla $b_i = 0$

  $a^{s_i} \equiv h_i h_j^{-1} \pmod{p}$ dla $b_i = 1$

- Dla każdego i, dla którego $b_i = 1$, Alicja oblicza i wysyła Bolkowi $z_i = (x - r_i) \pmod{p-1}$

- Bolek sprawdza czy $a^{z_i} \equiv b h_i^{-1} \pmod{p}$

15.6 Protokół Fiata-Shamira

- Bezpieczeństwo tego protokołu opiera się na trudności obliczeniowej pierwiastków kwadratowych modulo n, gdzie n jest iloczynem dwóch liczb pierwszych. Protokół ten wymaga udziału strony trzeciej, zaufanego arbitra — Trusted Authority (TA)
- TA wybiera dwie liczby pierwsze p i q, oblicza ich iloczyn $n = pq$
- Alicja wybiera losową liczbę względnie pierwszą z n, oblicza liczbę $v = s^2 \pmod{n}$ i rejestruje u TA v jako swój klucz publiczny
- TA udostępnia liczby n i v jako identyfikatory tożsamości Alicji

15.6 Protokół Fiata-Shamira

- Bezpieczeństwo tego protokołu opiera się na trudności obliczeniowej pierwiastków kwadratowych modulo n, gdzie n jest iloczynem dwóch liczb pierwszych. Protokół ten wymaga udziału strony trzeciej, zaufanego arbitra — Trusted Authority (TA)
- TA wybiera dwie liczby pierwsze p i q, oblicza ich iloczyn $n = pq$
- Alicja wybiera losową liczbę względnie pierwszą z n, oblicza liczbę $v = s^2 \pmod{n}$ i rejestruje u TA v jako swój klucz publiczny
- TA udostępnia liczby n i v jako identyfikatory tożsamości Alicji

15.6 Protokół Fiata-Shamira

- Bezpieczeństwo tego protokołu opiera się na trudności obliczeniowej pierwiastków kwadratowych modulo n, gdzie n jest iloczynem dwóch liczb pierwszych. Protokół ten wymaga udziału strony trzeciej, zaufanego arbitra — Trusted Authority (TA)
- TA wybiera dwie liczby pierwsze p i q, oblicza ich iloczyn $n = pq$
- Alicja wybiera losową liczbę względnie pierwszą z n, oblicza liczbę $v = s^2 \pmod{n}$ i rejestruje u TA v jako swój klucz publiczny
- TA udostępnia liczby n i v jako identyfikatory tożsamości Alicji

15.6 Protokół Fiata-Shamira

- Bezpieczeństwo tego protokołu opiera się na trudności obliczeniowej pierwiastków kwadratowych modulo n, gdzie n jest iloczynem dwóch liczb pierwszych. Protokół ten wymaga udziału strony trzeciej, zaufanego arbitra — Trusted Authority (TA)
- TA wybiera dwie liczby pierwsze p i q, oblicza ich iloczyn $n = pq$
- Alicja wybiera losową liczbę względnie pierwszą z n, oblicza liczbę $v = s^2 \pmod{n}$ i rejestruje u TA v jako swój klucz publiczny
- TA udostępnia liczby n i v jako identyfikatory tożsamości Alicji

15.6 Protokół Fiata-Shamira

- Bezpieczeństwo tego protokołu opiera się na trudności obliczeniowej pierwiastków kwadratowych modulo n, gdzie n jest iloczynem dwóch liczb pierwszych. Protokół ten wymaga udziału strony trzeciej, zaufanego arbitra — Trusted Authority (TA)
- TA wybiera dwie liczby pierwsze p i q, oblicza ich iloczyn $n = pq$
- Alicja wybiera losową liczbę względnie pierwszą z n, oblicza liczbę $v = s^2 \pmod{n}$ i rejestruje u TA v jako swój klucz publiczny
- TA udostępnia liczby n i v jako identyfikatory tożsamości Alicji

- Alicja wybiera losowo liczbę r względnie pierwszą z n, oblicza $x = r^2 \pmod{n}$ i wysyła x Bolkowi
- Bolek wysyła Alicji losowy bit b
- Alicja wysyła Bolkowi

  r jeśli $b = 0$

  $y = r \cdot s \pmod{n}$ jeśli $b = 1$
- Bolek sprawdza czy

  $x = r^2 \pmod{n}$ jeśli $b = 0$

  $y^2 = x \cdot v \pmod{n}$ jeśli $b = 1$

- Alicja wybiera losowo liczbę r względnie pierwszą z n, oblicza $x = r^2 \pmod{n}$ i wysyła x Bolkowi
- Bolek wysyła Alicji losowy bit b
- Alicja wysyła Bolkowi

  r jeśli $b = 0$

  $y = r \cdot s \pmod{n}$ jeśli $b = 1$
- Bolek sprawdza czy

  $x = r^2 \pmod{n}$ jeśli $b = 0$

  $y^2 = x \cdot v \pmod{n}$ jeśli $b = 1$

- Alicja wybiera losowo liczbę r względnie pierwszą z n, oblicza $x = r^2 \pmod{n}$ i wysyła x Bolkowi
- Bolek wysyła Alicji losowy bit b
- Alicja wysyła Bolkowi

  r jeśli $b = 0$

  $y = r \cdot s \pmod{n}$ jeśli $b = 1$
- Bolek sprawdza czy

  $x = r^2 \pmod{n}$ jeśli $b = 0$

  $y^2 = x \cdot v \pmod{n}$ jeśli $b = 1$

- Alicja wybiera losowo liczbę r względnie pierwszą z n, oblicza $x = r^2 \pmod{n}$ i wysyła x Bolkowi
- Bolek wysyła Alicji losowy bit b
- Alicja wysyła Bolkowi

  r jeśli $b = 0$

  $y = r \cdot s \pmod{n}$ jeśli $b = 1$
- Bolek sprawdza czy

  $x = r^2 \pmod{n}$ jeśli $b = 0$

  $y^2 = x \cdot v \pmod{n}$ jeśli $b = 1$

- Pierwsza równość dowodzi, że Alicja zna pierwiastek kwadratowy z x, druga natomiast dowodzi, że Alicja zna s. Protokół ten powtarza się t razy, wtedy prawdopodobieństwo oszustwa przez Alicję wynosi $1/2^t$.

15.7 Protokół Schnorra

- Protokół ten opiera się na problemie logarytmu dyskretnego.
- Protokół wykorzystuje certyfikaty wydawane przez TA
- W etapie wstępnym należy wybrać liczbę pierwszą p oraz drugą liczbę pierwszą q taką, że $q|p-1$. Liczby te powinny być dostatecznie duże (np. p długości 1024 bity a $q > 160$ bitów). Wybieramy także liczbę $b = g^{(p-1)/q}$, gdzie g jest generatorem $\mathbb{Z}_p$.
- Każda ze stron otrzymuje liczby $\{p, q, b\}$ oraz klucz publiczny pozwalający weryfikować podpisy TA. Ponadto należy wybrać parametr t ($t \geq 40, \quad 2^t < q$), który określa poziom bezpieczeństwa.

15.7 Protokół Schnorra

- Protokół ten opiera się na problemie logarytmu dyskretnego.
- Protokół wykorzystuje certyfikaty wydawane przez TA
- W etapie wstępnym należy wybrać liczbę pierwszą p oraz drugą liczbę pierwszą q taką, że $q|p-1$. Liczby te powinny być dostatecznie duże (np. p długości 1024 bity a $q > 160$ bitów). Wybieramy także liczbę $b = g^{(p-1)/q}$, gdzie g jest generatorem $\mathbb{Z}_p$.
- Każda ze stron otrzymuje liczby $\{p, q, b\}$ oraz klucz publiczny pozwalający weryfikować podpisy TA. Ponadto należy wybrać parametr t ($t \geq 40, \quad 2^t < q$), który określa poziom bezpieczeństwa.

15.7 Protokół Schnorra

- Protokół ten opiera się na problemie logarytmu dyskretnego.
- Protokół wykorzystuje certyfikaty wydawane przez TA
- W etapie wstępnym należy wybrać liczbę pierwszą p oraz drugą liczbę pierwszą q taką, że $q|p-1$. Liczby te powinny być dostatecznie duże (np. p długości 1024 bity a $q > 160$ bitów). Wybieramy także liczbę $b = g^{(p-1)/q}$, gdzie g jest generatorem $\mathbb{Z}_p$.
- Każda ze stron otrzymuje liczby $\{p, q, b\}$ oraz klucz publiczny pozwalający weryfikować podpisy TA. Ponadto należy wybrać parametr t ($t \geq 40, \quad 2^t < q$), który określa poziom bezpieczeństwa.

15.7 Protokół Schnorra

- Protokół ten opiera się na problemie logarytmu dyskretnego.
- Protokół wykorzystuje certyfikaty wydawane przez TA
- W etapie wstępnym należy wybrać liczbę pierwszą p oraz drugą liczbę pierwszą q taką, że $q|p-1$. Liczby te powinny być dostatecznie duże (np. p długości 1024 bity a $q > 160$ bitów). Wybieramy także liczbę $b = g^{(p-1)/q}$, gdzie g jest generatorem $\mathbb{Z}_p$.
- Każda ze stron otrzymuje liczby $\{p, q, b\}$ oraz klucz publiczny pozwalający weryfikować podpisy TA. Ponadto należy wybrać parametr t ($t \geq 40, \quad 2^t < q$), który określa poziom bezpieczeństwa.

15.7 Protokół Schnorra

- Protokół ten opiera się na problemie logarytmu dyskretnego.
- Protokół wykorzystuje certyfikaty wydawane przez TA
- W etapie wstępnym należy wybrać liczbę pierwszą p oraz drugą liczbę pierwszą q taką, że $q|p-1$. Liczby te powinny być dostatecznie duże (np. p długości 1024 bity a $q > 160$ bitów). Wybieramy także liczbę $b = g^{(p-1)/q}$, gdzie g jest generatorem $\mathbb{Z}_p$.
- Każda ze stron otrzymuje liczby $\{p, q, b\}$ oraz klucz publiczny pozwalający weryfikować podpisy TA. Ponadto należy wybrać parametr t ($t \geq 40, \quad 2^t < q$), który określa poziom bezpieczeństwa.

- TA ustala tożsamość Alicji w konwencjonalny sposób i przydziela jej identyfikator I_A
- Alicja wybiera losowo tajną liczbę a oraz oblicza $v = b^a \pmod{p}$ i rejestruje v u TA
- TA generuje podpis cyfrowy $S(I_A, v)$ oraz wydaje Alicji certyfikat $C = (I_A, v, S(I_A, v))$ wiążący I_A z v
- Alicja wybiera liczbę losową $r < q$ i oblicza $x = b^r \pmod{p}$
- Alicja przesyła Bolkowi certyfikat C oraz liczbę x
- Bolek sprawdza klucz publiczny Alicji sprawdzając podpis TA na certyfikacie

- TA ustala tożsamość Alicji w konwencjonalny sposób i przydziela jej identyfikator I_A
- Alicja wybiera losowo tajną liczbę a oraz oblicza $v = b^a \pmod{p}$ i rejestruje v u TA
- TA generuje podpis cyfrowy $S(I_A, v)$ oraz wydaje Alicji certyfikat $C = (I_A, v, S(I_A, v))$ wiążący I_A z v
- Alicja wybiera liczbę losową $r < q$ i oblicza $x = b^r \pmod{p}$
- Alicja przesyła Bolkowi certyfikat C oraz liczbę x
- Bolek sprawdza klucz publiczny Alicji sprawdzając podpis TA na certyfikacie

- TA ustala tożsamość Alicji w konwencjonalny sposób i przydziela jej identyfikator I_A
- Alicja wybiera losowo tajną liczbę a oraz oblicza $v = b^a \pmod p$ i rejestruje v u TA
- TA generuje podpis cyfrowy $S(I_A, v)$ oraz wydaje Alicji certyfikat $C = (I_A, v, S(I_A, v))$ wiążący I_A z v
- Alicja wybiera liczbę losową $r < q$ i oblicza $x = b^r \pmod p$
- Alicja przesyła Bolkowi certyfikat C oraz liczbę x
- Bolek sprawdza klucz publiczny Alicji sprawdzając podpis TA na certyfikacie

- TA ustala tożsamość Alicji w konwencjonalny sposób i przydziela jej identyfikator I_A
- Alicja wybiera losowo tajną liczbę a oraz oblicza $v = b^a \pmod{p}$ i rejestruje v u TA
- TA generuje podpis cyfrowy $S(I_A, v)$ oraz wydaje Alicji certyfikat $C = (I_A, v, S(I_A, v))$ wiążący I_A z v
- Alicja wybiera liczbę losową $r < q$ i oblicza $x = b^r \pmod{p}$
- Alicja przesyła Bolkowi certyfikat C oraz liczbę x
- Bolek sprawdza klucz publiczny Alicji sprawdzając podpis TA na certyfikacie

- TA ustala tożsamość Alicji w konwencjonalny sposób i przydziela jej identyfikator I_A
- Alicja wybiera losowo tajną liczbę a oraz oblicza $v = b^a \pmod p$ i rejestruje v u TA
- TA generuje podpis cyfrowy $S(I_A, v)$ oraz wydaje Alicji certyfikat $C = (I_A, v, S(I_A, v))$ wiążący I_A z v
- Alicja wybiera liczbę losową $r < q$ i oblicza $x = b^r \pmod p$
- Alicja przesyła Bolkowi certyfikat C oraz liczbę x
- Bolek sprawdza klucz publiczny Alicji sprawdzając podpis TA na certyfikacie

- TA ustala tożsamość Alicji w konwencjonalny sposób i przydziela jej identyfikator I_A
- Alicja wybiera losowo tajną liczbę a oraz oblicza $v = b^a \pmod{p}$ i rejestruje v u TA
- TA generuje podpis cyfrowy $S(I_A, v)$ oraz wydaje Alicji certyfikat $C = (I_A, v, S(I_A, v))$ wiążący I_A z v
- Alicja wybiera liczbę losową $r < q$ i oblicza $x = b^r \pmod{p}$
- Alicja przesyła Bolkowi certyfikat C oraz liczbę x
- Bolek sprawdza klucz publiczny Alicji sprawdzając podpis TA na certyfikacie

- Bolek wybiera losowo liczbę k $(1 \leq k \leq 2^t)$ i wysyła ją Alicji (challenge)
- Alicja sprawdza $1 \leq k \leq 2^t$ i wysyła Bolkowi $y = ak + r \pmod{q}$ (response)
- Bolek oblicza $z = b^y v^k \pmod{p}$ i jeśli $z = x$ uznaje, że tożsamość Alicji jest potwierdzona.

- Bolek wybiera losowo liczbę k $(1 \leq k \leq 2^t)$ i wysyła ją Alicji (challenge)
- Alicja sprawdza $1 \leq k \leq 2^t$ i wysyła Bolkowi $y = ak + r \pmod{q}$ (response)
- Bolek oblicza $z = b^y v^k \pmod{p}$ i jeśli $z = x$ uznaje, że tożsamość Alicji jest potwierdzona.

- Bolek wybiera losowo liczbę k $(1 \leq k \leq 2^t)$ i wysyła ją Alicji (challenge)
- Alicja sprawdza $1 \leq k \leq 2^t$ i wysyła Bolkowi $y = ak + r \pmod{q}$ (response)
- Bolek oblicza $z = b^y v^k \pmod{p}$ i jeśli $z = x$ uznaje, że tożsamość Alicji jest potwierdzona.