Podpis cyfrowy inaczej podpis elektroniczny to dane w postaci elektronicznej służące przy wymianie danych drogą elektroniczną do:

- identyfikacji osoby składającej podpis,
- potwierdzenia autentyczności dokumentu.

Podpis elektroniczny powinien spełniać następujące warunki:

- niepodrabialność - jest świadectwem świadomego podpisu autora pod dokumentem,
- autentyczność - przekonuje odbiorcę, że podpisujący rozważnie go podpisał,
- można go użyć tylko jeden raz - jest częścią dokumentu i nie można przenieść podpisu na inny dokument,
- kontrola spójności - pozwala na wykrycie każdej nieautoryzowanej zmiany w treści dokumentu,
- niezaprzeczalność - autor dokumentu nie może zaprzeczyć, że wysłał dokument i go podpisał.

Mechanizm podpisu cyfrowego nie daje pewności, że posługująca się nim osoba celowo nie fałszuje swojej tożsamości.

Urząd Certyfikacji (z ang. CA - Certyficate Authority) instytucja, która za pewną opłatą wydaje elektroniczne certyfikaty tożsamości dla osób lub instytucji. Zadaniem tego urzędu jest sprawdzenie tożsamości osoby, której wydaje certyfikat. Inne osoby mogą sprawdzić tożsamość właściciela certyfikatu opierając się na zaufaniu do urzędu, który go wydał.

Certyfikat cyfrowy posiada następujące informacje:

- unikalny numer seryjny,
- tożsamość urzędu certyfikacji wydającego certyfikat,
- identyfikator właściciela certyfikatu (imię i nazwisko osoby lub nazwa firmy, e-mail, itp.)
- klucz publiczny właściciela certyfikatu,
- podpis cyfrowy urzędu certyfikacji potwierdzający autentyczność certyfikatu.

Certyfikat cyfrowy może być wysłany w wiadomości razem z podpisem cyfrowym. W ten sposób osoba otrzymująca wiadomość otrzymuje również wszystkie informacje niezbędne do weryfikacji podpisu.

Podpisy elektroniczne korzystają z krytpografii asymetrycznej:

- Alicja generuje parę kluczy:
  - klucz prywatny służący do podpisywania dokumentu,
  - klucz publiczny służący do weryfikowania podpisu,

- Alicja chcąc podpisać dokument M:
  - generuje szyfrogram z M za pomocą klucza prywatnego,
  - publikuje szyfrogram jako podpis,
  - prezentuje oryginalny dokument wraz z podpisem,

- Bob chcąc przekonać się o podpisie Alicji deszyfruje szyfrogram za pomocą klucza publicznego. Bob jest pewny autentyczności nadawcy i danych.

Wady - podpis jest bardzo długi - co najmniej tak długi jak podpisywany dokument.

Zalety - gwarancja, że podpis nie może być przeniesiony na inny dokument.

Generowanie krótkich podpisów - zamiast podpisywać dokument M, podpisujemy wartość $h(M)$, gdzie h jest jednokierunkową funkcją skrótu.

Protokoły z wykorzystaniem jednokierunkowej funkcji skrótu

- Alicja oblicza wartość funkcji skrótu dla dokumentu, który ma podpisać,
- Alicja podpisuje skrót dokumentu szyfrując go za pomocą swojego klucza prywatnego,
- Alicja przesyła Bobowi dokument i podpisany skrót,
- Bob używa tej samej funkcji skrótu, oblicza jej wartość dla otrzymanego dokumentu, deszyfruje otrzymany skrót za pomocą klucza publicznego Alicji. Jeśli te dwie wartości się zgadzają, to podpis jest prawdziwy.

Zalety:

- podpis jest znacznie krótszy od dokumentu,
- można sprawdzić istnienie podpisu bez oglądania samego dokumentu.

Najważniejszymi kryptosystemami umożliwiającymi składanie podpisów cyfrowych są:

- RSA,
- ElGamala,
- DSA.

Najpopularniejsze standardy pozwalające na złożenie podpisu cyfrowego to:

- PGP (Pretty Good Privacy),
- X.509.

W systemie PGP stosowana jest sieć zaufania:

- każdy użytkownik PGP może sam wygenerować parę kluczy, korzystając z łatwo dostępnego oprogramowania,

- klucze publiczne publikowane są na specjalnych serwerach np. http://keyserver.pgp.com

- tożsamość właściciela klucza potwierdzają własnym podpisem cyfrowym inni użytkownicy PGP, którzy znają go osobiście. Może się zdarzyć, że klucz publiczny nieznanej nam osoby będzie podpisany przez znajomego, którego darzymy zaufaniem. Wówczas podpis znajomego złożony na kluczu uwierzytelnie jego posiadanie.

Uwaga 1. Istnieje ryzyko, że któryś z kluczy nie będzie zaufany.

W standardzie X.509 certyfikaty obsługiwane są przez infrastrukturę klucza publicznego (ang. public key infrastructure, PKI). Para kluczy oraz certyfikat wydawany jest przez Urząd Certyfikacji.

Zaletą standardu X.509 jest fakt, że obsługuje go zdecydowana większość programów pocztowych.

Podpisy cyfrowe tworzone za pomocą RSA

1. Alicja oblicza dla dokumentu M wartość $h(M)$, gdzie h jest ustaloną jednokierunkową funkcją skrótu,

2. wybór kluczy:

   (a) Alicja losowo wybiera dwie duże liczby pierwsze p, q i oblicza ich iloczyn $n = pq$,

   (b) Alicja losowo wybiera liczbę e taką, że NWD$(e, (p-1)(q-1)) = 1$,

   (c) za pomocą algorytmu Euklidesa Alicja znajduje d takie, że

   $$e \cdot d \equiv 1 \mod (p-1)(q-1),$$

   (d) $[e, n]$ jest wygenerowanym kluczem publicznym, $[d, n]$ jest kluczem prywatnym.

3. Alicja szyfruje $h(M)$ w następujący sposób:

$$E_{[d,n]}(h(M)) = h(M)^d \mod n.$$

Uwaga 2. W algorytmie RSA szyfrowane mogą być liczby $m < n$. Jeśli $m > n$, to m należy podzielić na bloki tej samej długości tak, aby każda liczba z bloku była mniejsza od n.

4. Podpisem jest $E_{[d,n]}(h(M))$.

5. Podpis jest akceptowany, gdy wartość jednokierunkowej funkcji skrótu jest równa wartości

$$D_{[e,n]}(C) = C^e \mod n,$$

gdzie C jest zaszyfrowaną wartością funkcji skrótu otrzymaną jako podpis.

Algorytm ElGamala dla podpisów

1. Alicja oblicza dla dokumentu M wartość $h(M)$, gdzie h jest ustaloną jednokierunkową funkcją skrótu.

2. Wybór kluczy:

   (a) Alicja wybiera liczbę pierwszą p,

   (b) Alicja wybiera dwie liczby losowe g, x takie, że $g, x < p$.

   (c) Alicja oblicza $y = g^x \mod p$.

   (d) y, g, p jest kluczem publicznym, x jest kluczem prywatnym.

3. Podpisywanie $h(M)$:

   (a) Alicja wybiera liczbę losową k względnie pierwszą z $p-1$,

   (b) Alicja oblicza $a = g^k \mod p$,

   (c) korzystając z rozszerzonego algorytmu Euklidesa Alicja oblicza liczbę b taką, że

$$h(M) = xa + kb \mod p - 1.$$

   (d) podpisem jest para liczb a, b. Losowa liczba k musi być trzymana w sekrecie.

4. Podpis jest akceptowany, gdy

$$y^a a^b \mod p = g^{h(M)} \mod p.$$

Algorytm DSA (Digital Signature Algorithm):

- algorytm podpisu cyfrowego zatwierdzony w 1994 roku przez NIST jako standard podpisu cyfrowego w USA (Digital Signature Standard)

- może być stosowany jedynie do tworzenia podpisów cyfrowych,

- bezpieczeństwo tego algorytmu opiera się na trudności obliczenia dyskretnego logarytmu.

1. Alicja oblicza wartość $h(M)$, gdzie h jest jednokierunkową funkcją skrótu opisaną przez algorytm SHA-1.

2. Wybór kluczy:

   (a) Alicja ustala długość klucza l spełniającą następujące warunki $512 \leq l \leq 1024$, $64|l$.

   (b) Alicja wybiera następujące liczby:

      i. liczbę pierwszą p taką, że $2^{l-1} < p < 2^l$,

      ii. liczbę pierwszą q taką, że $q|p-1$ i $2^{159} < q < 2^{160}$,

      Uwaga 3. Jeśli nie można wyznaczyć liczby q, to Alicja wybiera inną liczbę p.

      iii. liczbę h taką, że $1 < h < p-1$ i oblicza $g = h^{(p-1)/q} \mod p$,

      iv. liczbę x taką, że $0 < x < q$ i oblicza $y = g^x \mod p$.

   (c) Kluczem publicznym jest (p, q, g, y), zaś kluczem prywatnym jest x.

3. Podpisywanie $h(M)$:

   (a) Alicja wybiera losowo liczbę k taką, że $0 < k < q$,

   (b) Alicja oblicza:

$$\begin{aligned} r &= (g^k \mod p) \mod q, \\ s &= k^{-1}(h(M) + x \cdot r) \mod q. \end{aligned}$$

   (c) Podpisem jest para liczb (s, r).

4. Podpis jest akceptowany, gdy w wyniku obliczeń

$$\begin{aligned} w &= s^{-1} \mod q, \\ u_1 &= h(M) \cdot w \mod q, \\ u_2 &= r \cdot w \mod q, \\ v &= ((g^{u_1} \cdot y^{u_2}) \mod p) \mod q \end{aligned}$$

   otrzymamy $v = r$.

Poprawność konstrukcji

Zauważmy, że h i p są względnie pierwsze, więc z twierdzenia Eulera wynika, że $h^{p-1} \equiv 1 \mod p$, czyli $g^q = h^{p-1} = 1 \mod p$. Stąd $g^z = g^{z \mod q} \mod p$. Zatem prowadząc obliczenia modulo p otrzymujemy:

$$\begin{aligned} g^{u_1} \cdot y^{u_2} &= g^{u_1} \cdot (g^x)^{u_2} = g^{u_1} \cdot g^{x \cdot u_2} = g^{h(M) \cdot w} \cdot g^{x \cdot r \cdot w} = g^{(h(M)+x \cdot r) \cdot w} \\ &= g^{(h(M)+x \cdot r) \cdot s^{-1}} = g^{(h(M)+x \cdot r) \cdot k \cdot (h(M)+x\dot{r})^{-1}} = g^k \mod p. \end{aligned}$$

Wobec tego

$$v = ((g^{u_1} \cdot y^{u_2}) \mod p) \mod q = (g^k \mod p) \mod q = r.$$

Tajność klucza prywatnego to najbardziej newralgiczny element systemu z podpisem cyfrowym.

Bezpieczne przechowywanie klucza prywatnego zapewniają kryptograficzne karty mikroprocesorowe (SmartCard).
Karta taka zawiera:

- zakodowane dane właściciela podpisu,
- charakter prawny właściciela podpisu (osoba, instytucja),
- elektroniczny certyfikat wraz z kluczami publicznym i prywatnym.

Karta mikroprocesorowa

- jest zabezpieczona kodem PIN,
- komunikuje się z komputerem za pomocą czytnika,
- ma zaszyte algorytmy kryptograficzne, dlatego klucz zapisany w karcie nigdy nie wydostaje się na zewnątrz, ponieważ podpis cyfrowy generowany jest wewnątrz mikroprocesora karty.

Czytniki kart mogą być wewnętrzne lub zewnętrzne. Aby złożyć podpis cyfrowy należy włożyć kartę do czytnika i podać kod PIN.

Ślepe podpisy

Przypuśćmy, że Alicja pragnie, aby notariusz poświadczył, że jest ona w posiadaniu pewnego dokumentu. Alicja jednak nie chce pokazać tego dokumentu notariuszowi. W tym celu notariusz musi złożyć tzw. ślepy podpis:

- Alicja wkłada do koperty z listem kalkę i zakleja kopertę.
- Notariusz potwierdza na kopercie, że list został mu przedstawiony - składa na kopercie swój podpis.
- W odpowiednim momencie Alicja może otworzyć kopertę i pokazać podpis notariusza utworzony przez kalkę.

Do generowania ślepych podpisów cyfrowych wykorzystywany jest algorytm RSA.

Ślepy podpis z użyciem RSA

1. Notariusz używa klucza publicznego $[e, n]$ i klucza prywatnego $[d, n]$.

2. Alicja wybira liczbę losową k taką, że $0 < k < n$.

3. Alicja oblicza $t = M \cdot k^e \mod n$.

4. Alicja przesyła liczbę t notariuszowi.

5. Notariusz szyfruje t za pomocą swojego prywatnego klucza $s = t^d \mod n$.

6. Notariusz przesyła Alicji liczbę s.

7. Ponieważ $s = t^d = (M \cdot k^e)^d = M^d \cdot k^{ed} = M^d \cdot k \mod n$. Stąd $M^d = s \cdot k^{-1} \mod n$, więc Alicja łatwo może obliczyć M^d będące podpisaną wiadomością M.

Podpisy niezaprzeczalne:

- nie może być sprawdzony bez zgody autora,
- podpisujący nie może wyprzeć się swojego podpisu,
- w przypadku sfałszowania podpisu podpisujący ma możliwość udowodnienia fałszerstwa.

Algorytm składania niezaprzeczalnych podpisów oparty na dyskretnych logarytmach

Zakładamy, że Alicja ma zamiar podpisać dokument.

1. Generowanie klucza: Alicja generuje klucz prywatny x oraz klucz publiczny $\{y, g, p\}$ tak samo jak w algorytmie ElGamala.

2. Alicja oblicza dla dokumentu M liczbę $z = M^x \mod p$, która jest podpisem dla dokumentu M.

3. Weryfikacja podpisu:

   (a) Bob wybiera dwie liczby losowe r i s mniejsze od p i oblicza $w = z^r y^s \mod p$.

   (b) Bob przesyła Alicji obliczoną wartość w.

   (c) Alicja oblicza:

$$\begin{aligned} t &= x^{-1} \mod (p-1), \\ v &= w^t \mod p. \end{aligned}$$

4. Alicja przesyła Bobowi obliczone v.

5. Bob sprawdza, czy $v = M^r g^s \mod p$.

Uzasadnienie

Zauważmy, że

$$v = w^t = z^{rt} y^{st} = M^{xrt} g^{xst} = (M^{xt})^r (g^{xt})^s.$$

Z definicji liczby t wynika, że $M^{xt} = M \mod p$ i $g^{xt} = g \mod p$.
Stąd $v = M^r g^s \mod p$.

Uwierzytelnianie jest jednym z kluczowych zadań w zapewnieniu bezpieczeństwa w systemach komputerowych. Służy do

- zapewnienia dostępu do systemu i zasobów tylko osobom do tego uprawnionym,

- sprawdzania tożsamości użytkownika.

Protokół challenge and response z kluczem tajnym

- Alicja i Bob ustalają na początku funkcję jednokierunkową f, którą będą używać oraz wartość klucza tajnego K.

- Alicja komunikuje się z Bobem przedstawiając się.

- Bob generuje losowy ciąg r i przesyła go Alicji.

- Alicja oblicza $f(K, r)$ i przesyła wynik do Boba.

- Bob oblicza także $f(K, r)$. Jeśli wynik zgadza się z wynikiem przesłanym przez Alicję, to tożsamość Alicji zostnie potwierdzona.

Uwaga 4. Funkcja f stosowana w powyższym protokole musi posiadać następującą własność:

dla $k \neq k'$ z dużym prawdopodobieństwem zachodzi $f(k, r) \neq f(k', r)$.

Protokół challenge and response z kluczem publicznym

- Alicja komunikuje się z Bobem przedstawiając się.
- Bob generuje losowy ciąg r i przesyła go Alicji.
- Alicja szyfruje r za pomocą jej prywatnego klucza i wysyła szyfrogram Bobowi.
- Bob deszyfruje za pomocą klucza publicznego otrzymaną wiadomość. Jeśli otrzymuje ten sam ciąg losowy, który wysłał Alicji, to tożsamość Alicji jest potwierdzona.

Dowody z wiedzą zerową

Alicja ma przekonać Boba, że zna pewien sekret nie zdradzając żadnych informacji o sekrecie.

Przykład 5. Alicja i Bob znajdują się przed labiryntem, w którym znajdują się drzwi zaopatrzone w zamek szyfrowy. Alicja twierdzi, że zna kod potrzebny do otwarcia drzwi. Dowód będzie przebiegał w k fazach. Pojedyńcza faza wygląda następująco:

- Alicja i Bob stoją przed wejściem do labiryntu.
- Alicja wchodzi do labiryntu i idzie w prawo lub w lewo dochodząc do drzwi zamykających przejście.
- Bob nie widzi, w którą stronę poszła Alicja - korytarz w labiryncie jest załamany.
- Bob idzie do rozwidlenia korytarza, rzuca monetą i zgodnie z wynikiem rzutu nakazuje Alicji przyjść z prawej lub lewej strony.

- Alicja wykonuje polecenie Boba przechodząc przez drzwi jeśli to konieczne.

Uwaga 6. Prawdopodobieństwo, że Alicja wykona polecenia Boba nie potrafiąc otworzyć drzwi wynosi $1/2^k$. Jeśli k jest dostatecznie duże, to prawdopodobieństwo to jest bardzo małe.

Dowód z wiedzą zerową dla dyskretnego logarytmu

Niech
p będzie liczbą pierwszą,
x liczbą względnie pierwszą z p,
$a^x = b \mod p$.
Alicja i Bob znają p, a, b.
Alicja zna liczbę x i chce przekonać o tym Boba.

1. Alicja generuje t losowych liczb $r_1, \ldots, r_t$.

2. Alicja oblicza $h_1 = a^{r_1} \mod p, \ldots, h_t = a^{r_t} \mod p$ i przedstawia te liczby Bobowi.

3. Alicja i Bob wspólnie rzucają t razy monetą i generują w ten sposób t losowych bitów $b_1, \ldots, b_t$.

4. Niech $j := \max\{k \leq t : b_k = 1\}$. Dla $i \leq t$ Alicja wylicza i wysyła Bobowi następujące liczby:

$$y_i = \begin{cases} r_i, & \text{gdy } b_i = 0, \\ r_i - r_j, & \text{gdy } b_i = 1. \end{cases}$$

5. Bob sprawdza dla każdego $i \leq t$, czy otrzymane liczby y_i są poprawnie zbudowane:

$$a^{y_i} = \begin{cases} h_i, & \text{gdy } b_i = 0, \\ h_i/h_j, & \text{gdy } b_i = 1. \end{cases}$$

6. Dla każdego i, dla którego $b_i = 1$, Alicja wysyła Bobowi liczbę $z_i = x - r_i \mod p$.

7. Bob sprawdza, czy $a^{z_i} = b/h_i \mod p$ dla tych liczb i, dla których $b_i = 1$.