7. Podpisy cyfrowe - uwierzytelnianie wiadomości

7.1 Podpisy cyfrowe – uwagi wstępne

Podpis cyfrowy (ang *digital signature*) nazywamy również podpisem elektronicznym (ang. *electronic signature*). Podpis cyfrowy jest odpowiednikiem zwykłego podpisu piórem jaki składamy pod dokumentem. Podpisy cyfrowe z reguły wymagają użycia systemu kryptograficznego z kluczem publicznym lub są specjalnymi algorytmami.

1. Trzy zasadnicze cechy podpisów (podpisów w ogóle, a w szczególności podpisów cyfrowych):

   I. Jedynie osoba X może utworzyć podpis osoby X (czyli podrobienie podpisu powinno być niewykonalne i nie mogą istnieć 2 osoby o jednakowym podpisie).

   II. Powinno się dać jednoznacznie stwierdzić czy podpis osoby X został złożony pod danym dokumentem *m* (czyli podpis powinien być weryfikowalny).

   III. Kopiowanie podpisu z jednego dokumentu na drugi powinno być niewykonalne (czyli podpis cyfrowy powinien być nierozerwalnie związany z dokumentem podpisywanym).

UWAGA 1. Podpisy cyfrowe mają trzy powyższe cechy i gwarantują dużo wyższy poziom bezpieczeństwa niż podpisy tradycyjne.

UWAGA 2. Sytuacja prawna. Od 1999 roku podpisy cyfrowe są uznanym prawnie sposobem podpisywania dokumentów w Unii Europejskiej a od 2000 roku w Stanach Zjednoczonych. Skończyły się już prace legislacyjne zmierzające do wprowadzenia analogicznych regulacji prawnych w Polsce. Ustawa o podpisie cyfrowym weszła w życie w sierpniu 2002 roku. Zgodnie z unormowaniami tej ustawy podpis elektroniczny złożony pod dokumentem wysłany przez Internet np. do Urzędu Finansowego ma takie same skutki prawne jak podpis złożony na papierze.

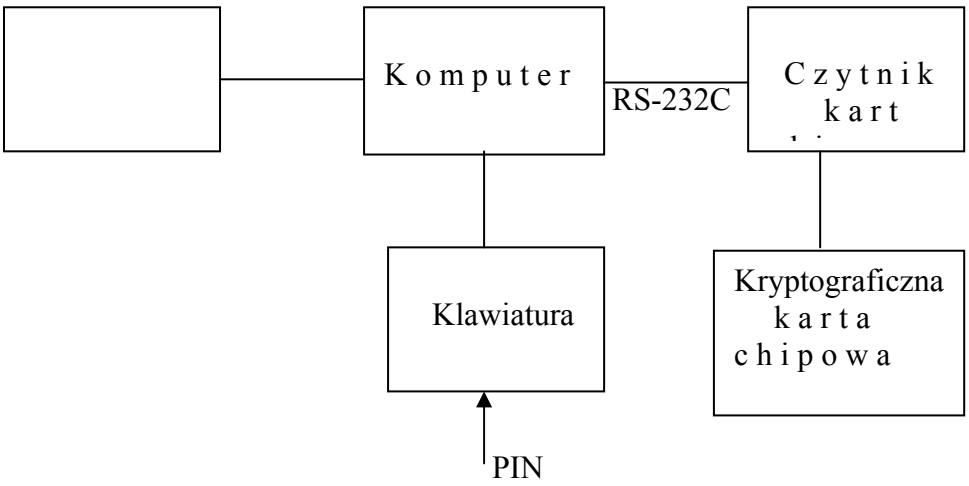
UWAGA 3. Typowe sfery zastosowań podpisów cyfrowych to: e-banking (w tym pieniądze elektroniczne czyli e-cash), e-commerce czyli handel elektroniczny oraz EDI (Electronic Data Interchange) czyli elektroniczna wymiana danych i dokumentów. Podpisy cyfrowe stały się obecnie standardem w obrocie bezgotówkowym, realizacjach płatności i ogólnie rzecz biorąc operacjach bankowych.

Elektroniczna wymiana dokumentów. Szczególnie ważną rolę odgrywają podpisy cyfrowe w elektronicznej wymianie danych i dokumentów tzw. EDI (Electronic Data Interchange lub Electronic Document Interchange). Dokument zawsze obok treści wymaga jeszcze 2 elementów: podpisu i daty. Dokument cyfrowy w postaci pliku można za pomocą odpowiedniego algorytmu podpisu cyfrowego zaopatrzyć i w podpis cyfrowy i w datę-stempel. Zaopatrywanie dokumentu w weryfikowalną i związaną z dokumentem datę utworzenia tego dokumentu nosi nazwę stemplowania czasem. Stemplowanie czasem nazywamy często w żargonie informatycznym „time stamping'iem".

EDI ma kilka standardów np. UN/EDIFACT, XML, ANSI X12. Szeroko wykorzystywany jest w EDI język opisu dokumentów XML.

Standardy EDI używane są rutynowo w handlu, usługach, finansach i zarządzaniu przedsiębiorstwem (przesyłanie faktur, zamówień, umów, dokumentów celnych, zleceń zakupu, sprawozdań itp.). Coraz częściej z EDI korzysta również administracja państwowa, służba zdrowia i służby logistyczne.

Przesyłanie i przechowywanie dokumentów w postaci elektronicznej (mówimy też o automatyzacji obiegu dokumentów) jest oczywiście tańsze i szybsze niż tradycyjne, a dzięki podpisom cyfrowym i szyfrowaniu jest również całkowicie bezpieczne. Znacznie łatwiejsze i szybsze jest ponadto wyszukiwanie i przetwarzanie dokumentów w postaci elektronicznej.



Rys 1. Typowy zestaw umożliwiający składanie podpisów cyfrowych

7.2 Algorytm podpisów cyfrowych RSA

Algorytm podpisów cyfrowych wykorzystujący system kryptograficzny z kluczem publicznym

Najprostszy sposób realizacji podpisów cyfrowych to realizacja za pomocą systemu kryptograficznego z kluczem publicznym. Zakładamy, że znane są ogólnie parametry tego systemu kryptograficznego (oczywiście poza kluczem prywatnym, którym dysponuje tylko osoba podpisująca dokument Alicja). Alicja dysponując kluczami: prywatnym k_e i publicznym k_d chce podpisać dokument *m*.

1. Alicja ma tajny, prywatny klucz k_e i ogłasza klucz publiczny k_d.

2. Podpisanie wiadomości *m*, (może to być list czy dokument) polega na obliczaniu kryptogramu $E(k_e,m)$ i to jest podpis cyfrowy.

3. Jako podpisany dokument jest przedstawiany oryginalny dokument *m* wraz z kryptogramem $E(k_e,m)$ czyli para uporządkowana $(m,E(k_e,m))$.

4. Osoba pragnąca sprawdzić, czy rzeczywiście dokument *m* jest podpisany przez Alicję deszyfruje kryptogram za pomocą klucza publicznego k_d. Jeśli otrzyma tekst *m* to oznacza to, że *m* jest podpisany przez Alicję.

Uwaga Wykorzystaliśmy w powyższym schemacie klucz prywatny k_e do szyfrowania dokumentu podpisywanego. Nie każdy system z kluczem publicznym dopuszcza szyfrowanie kluczem prywatnym a to jest istota rzeczy w powyższym schemacie podpisów.

Oczywistą zaletą tego sposobu podpisywania dokumentów jest to, że podpis nie może być przeniesiony na inny dokument. Zmiana dokumentu *m* pociąga za sobą zmianę podpisu $E(k_e,m)$ funkcja $E(\cdot,k_e):V_1^* \to V_2^*$ jest bowiem różnowartościowa.

Przy opisie systemu podpisów cyfrowych zwracamy uwagę na trzy rzeczy:

I. Ustalenie parametrów systemu podpisów cyfrowych.

II. Sposób generowania podpisu a dokładniej algorytm generowania podpisu. Osoba podpisująca dokument dokonuje obliczeń, w trakcie których powstaje słowo nad alfabetem V_2 (w praktyce najczęściej ciąg bitów) będący podpisem konkretnego dokumentu *m*.

III. Sposób weryfikacji podpisu a dokładniej algorytm weryfikacji podpisu. Są to obliczenia dokonywane na podpisie i wiadomości przez osobę, która pragnie się przekonać o autentyczności podpisu i autentyczności dokumentu. Weryfikacja jest testem, który powinny przechodzić jedynie poprawnie utworzone podpisy.

Tworzenie krótkich podpisów

Wadą powyżej opisanej metody uzyskiwania podpisu cyfrowego wiadomości *m* jest długość podpisu. Podpis ma na ogół zbliżoną długość do długości wiadomości jawnej. Oczywiście prostym środkiem zaradczym jest szyfrowanie kluczem prywatnym tylko wartości funkcji skrótu dla dokumentu *m*.

Podpisujemy zamiast dokumentu *m* wartość funkcji skrótu (czyli inaczej wartość funkcji hashującej). Zastosowanie funkcji skrótu $h : V_1^* \to V_2^k$, (gdzie $k \in N$ jest ustaloną liczbą), ma tę zaletę, że podpis można przedstawić do sprawdzenia nie zdradzając treści dokumentu *m*.

Podpisy cyfrowe RSA (bez funkcji skrótu i z funkcją skrótu)

Podpisy cyfrowe RSA bez funkcji skrótu

Podpisy cyfrowe RSA bez funkcji skrótu niczym nie różnią się od przedstawionego wyżej ogólnego schematu wykorzystującego system kryptograficzny z kluczem publicznym

Podpisy cyfrowe RSA z funkcją skrótu

Wstępnie ustalamy funkcję skrótu $h : V_1^* \to V_1^k$ oraz parametry systemu kryptograficznego *RSA* (por. rozdział o szyfrach z kluczem publicznym). Parametrami są: liczba bezkwadratowa n oraz $k_e, k_d \in Z_{\varphi(n)}$ (czyli klucz prywatny k_e i publiczny k_d). Podajemy do publicznej wiadomości h, n, k_d klucz k_e pozostaje tajny. Przyjmuje się, że

1. Dla dokumentu m obliczana jest wstępnie wartość *h(m)* funkcji hashującej.

2. Alicja składająca podpis pod dokumentem *m* szyfruje *h(m)* za pomocą swojego tajnego klucza prywatnego k_e (zakładamy, że tylko ona dysponuje tym kluczem) używając algorytmu RSA. Uzyskany kryptogram $k_e(h(m))$ jest podpisem Alicji pod dokumentem *m*.

3. Sprawdzenie podpisu Alicji przez Boba sprowadza się do deszyfracji podpisu $k_e(h(m))$ publicznym ogólnie znanym kluczem Alicji k_d tzn. do obliczenia: $k_d(k_e(h(m)))$. Jeśli $k_d(k_e(h(m)))=h(m)$ to oznacza to, że dokument m podpisała Alicja.

UWAGA. Bobowi, który weryfikuje autentyczność podpisu i dokumentu może być przedstawiona para uporządkowana $(m, k_e(h(m)))$ bądź para uporządkowana $(h(m), k_e(h(m)))$. W drugim przypadku stwierdza on autentyczność dokumentu *m* nie zapoznając się z *m*. Znajomość treści dokumentu *m* nie jest Bobowi potrzebna. Bob sprawdza jedynie czy $k_d(k_e h(m)) = h(m)$

7.3 Algorytm podpisów cyfrowych ElGamala

1. Wybór parametrów algorytmu.

Wybieramy dużą liczbę pierwszą p (np. o 100-150 cyfrach dziesiętnych). Obliczenia będziemy prowadzić w ciele F_p. Liczbę pierwszą p wybieramy tak, by problem logarytmu dyskretnego w grupie multiplikatywnej F_p^* był praktycznie nierozwiązalny. Wybieramy dowolny generator g grupy multiplikatywnej F_p^*. Z twierdzenia mówiącego, że grupa multiplikatywna każdego ciała skończonego jest cykliczna wynika, że taki generator g istnieje. Zauważmy, że grupa multiplikatywna F_p^* jest izomorficzna z grupą addytywną pierścienia Z_{p-1} (izomorfizmem jest tu np. odwzorowanie $f : Z_{p-1} \ni n \to f(n) = g^n \in F_p$)

Alicja wybiera (to ona będzie podpisywać dokumenty swoim prywatnym kluczem tajnym) losowo liczbę $x \in <2, p-2>$. Liczba x będzie tajemnicą Alicji, cechą wyróżniającą Alicję, jej prywatnym kluczem tajnym używanym do podpisywania dokumentów.

Uwaga. Wybór wartości x=1 lub x=p-1 ujawnia natychmiast x przy znajomości parametrów jawnych systemu więc losujemy x ze zbioru $<2, p-2>$.

Alicja oblicza $y = g^x$ w ciele F_p i ujawnia g, p, y jako klucz publiczny. Teraz widać dlaczego $x = 1$ i $x = p-1$ zdradzają x. Dla x=1 mamy bowiem $y = g$ i stąd zgadujemy natychmiast x. Musi ono być równe 1. Dla x=p-1 mamy z kolei na mocy małego twierdzenia Fermata $g^{p-1} = 1 = y$ i stąd, że y=1, zgadujemy x=p-1.

2. Podpisywanie dokumentu. (czynności Alicji przy podpisywaniu dokumentu m)

Dokument to dla nas liczba $m \in Z_{p-1}$. Podpisem wiadomości jawnej m przedstawionym przez Alicję będzie para uporządkowana $(a,b) \in Z_p \times Z_{p-1}$. Liczby a i b oblicza Alicja następująco.

1. **Obliczenie *a***. Alicja wybiera losowo $k \in Z_{p-1}$ takie, że NWD(k,p-1) =1 .Będziemy obliczali odwrotność k czyli k^{-1} w pierścieniu Z_{p-1} stąd założenie NWD(k,p-1) =1. Dobrych tzn. odwracalnych k jest $\varphi(p-1)$. Obliczamy teraz wartość $a = g^k$ w ciele F_p.

Oczywiście a jest losowe. Zauważmy, że przy przyjętych założeniach $a \neq 1$, $a \neq 0$, $a \neq p-1$ (czyli $a \neq -1$). Zatem $[a]_{p-1}$ jest odwracalne w Z_{p-1}.

2. Obliczenie *b*. Ponieważ NWD(k,p-1) =1 możliwe jest obliczenie $t = k^{-1}$ w pierścieniu Z_{p-1} . Alicja oblicza $t = k^{-1}$ w pierścieniu Z_{p-1} tzn. znajduje takie $t \in Z_{p-1}$,że $k \otimes_{p-1} t = 1$ w Z_{p-1} (t może być wyznaczone z twierdzenia Eulera lub rozszerzonego algorytmu Euklidesa). Mając t obliczamy liczbę b w pierścieniu Z_{p-1} następująco

$$b = t \otimes_{p-1} (m -_{p-1} x \otimes_{p-1} [a]_{p-1})$$

UWAGA. $m = k \otimes_{p-1} b \oplus_{p-1} x \otimes_{p-1} [a]_{p-1}$

3 Weryfikacja podpisu. Osoba weryfikująca podpis otrzymuje parę uporządkowaną (wiadomość jawna, podpis) czyli $(m,(a,b)) \in Z_{p-1} \times (Z_p \times Z_{p-1})$ i sprawdza czy

$$y^a \otimes_p a^b = g^m \text{ (podnoszenie do potęgi w } F_p \text{).}$$

Jeśli równość zachodzi, to akceptuje podpis, jeśli nie zachodzi to nie akceptuje.

Fakt Jeśli podpis $(a,b) \in Z_p \times Z_{p-1}$ został utworzony dla wiadomości m, to równość weryfikacyjna $y^a \otimes_p a^b = g^m$ (podnoszenie do potęgi w F_p).jest spełniona
Dowód. Istotnie

$$y^a \otimes_p a^b = g^{x \cdot a} \otimes_p g^{k \cdot b} = g^{ax+kb} = g^{[a \cdot x + kb]_{p-1}} = g^{[a]_{p-1} \otimes_{p-1} x \oplus_{p-1} k \otimes_{p-1} b} = g^m$$

■

7.4 Algorytm podpisów cyfrowych DSS

Skróty DSS (Digital Signature Standard) i DSA (Digital Signature Algorithm) oznaczają ten sam algorytm podpisów cyfrowych.

Algorytm DSS jest standardem podpisów cyfrowych zaproponowanych w 1991 roku przez amerykański National Institute of Standards and Technology (NIST). Jest to standardowa metoda cyfrowego podpisywania korespondencji stosowana przez organizacje rządowe i handlowe na całym świecie.

Bezpieczeństwo DSS wynika z trudności rozwiązania problemu logarytmu dyskretnego w skończonych ciałach prostych F_q przy dużym q.

Algorytm podpisów cyfrowych DSS jest następujący:

I. Wybór parametrów algorytmu

1. Wybieramy losowo liczbę pierwszą q o długości ≈160 bitów (stosuje się do tego generator liczb losowych i algorytm testujący pierwszość liczby).

2. Wybieramy 2-gą liczbę pierwszą $p>q$ o długości $512 \le L \le 1024$ bitów taką, że $p \equiv 1 (\text{mod } q)$ czyli $p-1 \equiv 0 \ (\text{mod } q)$ co oznacza , że p-1 jest podzielne przez q . (przyjmujemy na ogół, że $64 \mid L$, czyli L jest wielokrotnością 64 ale nie jest to istotne przy dowodzie poprawności DSS).

UWAGA. Takie p i q nietrudno znaleźć wystarczy najpierw wybrać liczbę pierwszą p a potem jako q przyjąć dzielnik p-1.

3. Wybieramy generator g_0 podgrupy cyklicznej rzędu q (chodzi o podgrupę grupy multiplikatywnej $F_p^* = \{1,2,...p-1\}$). Z twierdzenia o strukturze grupy multiplikatywnej wynika, że odpowiednia podgrupa rzędu q istnieje. W celu określenia tej podgrupy obliczamy dla losowo wybranego g_0 liczbę $g = g_0^{(p-1)/q} \ (\text{mod } p)$. Jeśli $g \neq 1$ to g_0 jest generatorem grupy F_p^* a g generatorem podgrupy cyklicznej rzędu q . Jeśli $g = 1$ to wybieramy inne g_0 .

4. Wybieramy losową liczbę $x \in F_q \setminus \{0,1,q-1\}$, będzie to klucz prywatny, i ujawniamy $y = g^x$ (podnosimy do potęgi w F_p), będzie to klucz publiczny. (Dla x=0,1,q-1 bardzo łatwo zgadnąć klucz prywatny x mając dany klucz publiczny y więc liczb 0,1,q-1 nie wybieramy).

UWAGA. x- jest kluczem prywatnym, y- jest kluczem publicznym

Liczby p, q, g są ogólnie znanymi parametrami podanymi do publicznej wiadomości.

II. Tworzenie podpisu cyfrowego.

1.Wybieramy losowo liczbę $k \in F_q \setminus \{0,1,q-1\}$.

2.Obliczamy 2 liczby *r* i *s* następująco

$$r = [g^k]_q \text{ (podnoszone do potęgi w } F_p),$$

Jeśli k jest wybierane losowo to oczywiście r jest również losowe!

$$s = k^{-1} \otimes_q (SHA(m) \oplus_q x \otimes_q r$$

gdzie $SHA(m)$ jest funkcją skrótu wiadomości podpisywanej m .

Podajemy parę liczb $(s,r) \in F_q \times F_q$ jako podpis wiadomości jawnej *m*.

UWAGA. Zauważmy, że (*s,r*) tworzymy z *m*, tajnego klucza prywatnego *x* oraz losowej liczby *k*.

II Sprawdzanie podpisu cyfrowego.

Jeśli chcemy sprawdzić podpis cyfrowy, to robimy to tak:

1. Obliczamy $w = s^{-1}$ (odwrotność w F_q).
2. Obliczamy $u_1 = SHA(m) \otimes_q w$ (w F_q).
3. $u_2 = r \otimes_q w$
4. $v = [g^{u_1} \otimes_p y^{u_2}]_q$ (podnoszone do potęgi w F_p)
5. Jeśli *v=r* to podpis jest prawidłowy.

Uwaga. Sprawdzający podpis dysponuje więc takimi danymi: $(m,(s,r))$

Uzasadnienie poprawności

Z twierdzenia Eulera: $g^q = (h^{(p-1)/q})^q = h^{p-1} = 1$ (potęgowane w F_p) zatem $g^z = g^{[z]_q}$ i dalej potęgując w F_p mamy:

$$g^{u_1} y^{u_2} = g^{SHA(m)\otimes_q w} \otimes_p g^{x\otimes_q r\otimes_q w} = g^{SHA(m)\otimes_q w\oplus_q x\otimes_q r\otimes_q w} = g^{(SHA(m)\oplus_q x\otimes_q r)\otimes_q w} =$$
$$= g^{(SHA(m)\oplus_q x\otimes_q r)\otimes_q k\otimes_q (SHA(m)\oplus_q x\otimes_q r)^{-1}} = g^k$$

Ostatecznie więc mamy: $[g^{u_1} \otimes_p y^{u_2}]_q = [g^k]_q = r$

UWAGA 1. Jeśli stosujemy liczbę *q* o długości 160 bitów, to podpis cyfrowy czyli parę liczb (*s, r*) zapiszemy na 320 bitach.

UWAGA 2. Zasadniczą rolę w uzasadnianiu poprawności DSS odgrywa równość $g^z = g^{[z]_q}$ w ciele F_p. (podnoszenie do potęgi w ciele F_p).

UWAGA 3. Jeśli znamy tajną wartość *x*, to potrafimy podpisać się za nadawcę wiadomości przesyłając mu naszą wiadomość *m*. Bezpośredni atak na DSS polegałby na obliczeniu logarytmu dyskretnego w F_p z liczby *y*.

Ponieważ *g* nie jest generatorem grupy F_p^*, zadanie obliczenia logarytmu dyskretnego jest nieco prostsze (przy algorytmie pomnóż, sprawdź) niż w przypadku gdy *g* jest generatorem F_p^*.

7.4 Algorytm podpisów ślepych

Cel.

Chcemy przedstawić dokument *m* notariuszowi nie ujawniając mu treści tego dokumentu. (*m* to może być np. testament, treść patentu). Chcemy to zrobić w ten sposób, by każdy w momencie zaistnienia kontrowersji co do treści lub daty złożenia dokumentu mógł sprawdzić czy ten dokument istotnie przeszedł przez ręce notariusza.

Ustalenia wstępne.

Notariusz używa szyfru RSA i ma klucz publiczny (*e, n*) i prywatny (*d, n*) służący do podpisywania dokumentów..

Algorytm podpisów ślepych

1.Alicja wybiera liczbę losową $k \in Z_n / \{0\}$ i szyfruje k kluczem publicznym notariusza przesyłając notariuszowi liczbę

$$t = m \otimes_n k^e \text{ (potęgowanie w } Z_n \text{)}$$

2.Notariusz szyfruje *t* swoim kluczem prywatnym odsyłając Alicji liczbę

$$s = t^d \text{ (potęgowanie w } Z_n \text{).}$$

3.Zauważmy, że $s = t^d = (m \otimes_n k^e)^d = m^d \otimes_n k^{ed} = m^d \otimes_n k$. Ponieważ Alicja zna *k*, może obliczyć m^d mnożąc *s* przez k^{-1} w pierścieniu Z_n.

UWAGA. Finalnie Alicja ma dowód przedstawiając m i m^d, że dokument przeszedł przez ręce notariusza czyli, że został przez niego podpisany, co nie znaczy, że notariusz zapoznał się z jego treścią.

Literatura

[1] A. Menezes, P. Oorschot, S. Vanstone; Handbook of Applled Cryptography; CRC Press Inc., 1997. (treść jest na stronie: http://cacr.math.uwaterloo.ca/hac)

[2] J.Stokłosa, T.Bilski,T.Pankowski; Bezpieczeństwo danych w systemach informatycznych; PWN, Warszawa 2001.

[3] N.Koblitz; Algebraiczne aspekty kryptografii; WNT, Warszawa 2000.

[4] N.Koblitz; A Course in Number Theory and Cryptography; Springer Verlag, New York 1994. (jest przekład polski p.t. Wykład z teorii liczb i kryptografii; WNT, Warszawa 1995.)

[5] M.Kutyłowski, W.Strothmann; Kryptografia, teoria i praktyka zabezpieczania systemów komputerowych; Oficyna Wydawnicza Read Me, Warszawa 2001.

[6] B.Schneier; Kryptografia dla praktyków; WNT, 2002.

[7] J.Gawinecki, J.Szmidt; Zastosowanie ciał skończonych i krzywych eliptycznych w kryptografii; Wojskowa Akademia Techniczna, Warszawa 2003.

[8] T.HCormen, C.E.Leiserson, R.L.Rivest, C.Stein; Introduction to Algorithms; MIT 2001. (jest przekład polski p.t. Wprowadzenie do algorytmów; WNT, 2004.)

[9] W.Stallings; Ochrona danych w sieci i intersieci; WNT; Warszawa 1997.