Algorytm podpisu cyfrowego

Algorytm podpisu cyfrowego (**DSA)** to kryptosystem klucza publicznego i federalny standard przetwarzania informacji dla podpisów cyfrowych, oparty na matematycznej koncepcji potęgowania modułowego i problemie logarytmu dyskretnego. W kryptosystemie klucza publicznego generowane są dwa klucze: dane mogą być szyfrowane tylko za pomocą klucza publicznego, a zaszyfrowane dane można odszyfrować tylko za pomocą klucza prywatnego. DSA jest wariantem schematów podpisu Schnorra i ElGamal. [1]: 486

W 1991 r. Narodowy Instytut Standaryzacji i Technologii (NIST) zaproponował DSA do stosowania w swoim standardzie podpisu cyfrowego (DSS), a w 1994 r. przyjął go jako FIPS 186. [2] Opublikowano pięć poprawek do wstępnej specyfikacji. Najnowsza specyfikacja to: FIPS 186-5 (https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf) od lutego 2023 r. [3] DSA jest opatentowany, ale NIST udostępnił ten patent na całym świecie bez opłat licencyjnych. Specyfikacja FIPS 186-5 (https://doi.org/10.6028/NIST.FIPS.186-5) wskazuje, że DSA nie będzie już zatwierdzana do generowania podpisów cyfrowych, ale może być używana do weryfikacji podpisów wygenerowanych przed datą wdrożenia tego standardu.

Przegląd

DSA działa w ramach kryptosystemów klucza publicznego i opiera się na algebraicznych własnościach potęgowania modularnego, wraz z problemem logarytmu dyskretnego, który jest uważany za obliczeniowo nierozwiązywalny. Algorytm wykorzystuje parę kluczy składającą się z klucza publicznego i klucza prywatnego. Klucz prywatny służy do generowania podpisu cyfrowego dla wiadomości, a taki podpis można zweryfikować za pomocą odpowiedniego klucza publicznego osoby podpisującej. Podpis cyfrowy zapewnia uwierzytelnianie wiadomości (odbiorca może zweryfikować pochodzenie wiadomości), integralność (odbiorca może sprawdzić, czy wiadomość nie została zmodyfikowana od czasu jej podpisania) i niezaprzeczalność (nadawca nie może fałszywie twierdzić, że nie podpisał wiadomości).

Historia

W 1982 r. rząd Stanów Zjednoczonych zwrócił się do rządu USA z prośbą o wprowadzenie standardu podpisu klucza publicznego. W sierpniu 1991 roku National Institute of Standards and Technology (NIST) zaproponował DSA do wykorzystania w swoim standardzie podpisu cyfrowego (DSS). Początkowo spotkało się to ze znaczną krytyką, zwłaszcza ze strony firm programistycznych, które już zainwestowały wysiłek w rozwój oprogramowania do podpisu cyfrowego opartego na kryptosystemie RSA. [1]: 484 Niemniej jednak NIST przyjął DSA jako standard federalny (FIPS 186) w 1994 roku. Opublikowano pięć wersji początkowej specyfikacji: FIPS 186-1 w 1998 r.[4], FIPS 186-2 w 2000 r.[5], FIPS 186-3 w 2009 r.[6], FIPS 186-4 w 2013 r.[3] i FIPS 186-5 w 2023 r. [7] Standard FIPS 186-5 zabrania podpisywania się za pomocą DSA, jednocześnie zezwalając na weryfikację podpisów wygenerowanych przed datą wdrożenia standardu jako dokumentu. Ma on zostać zastąpiony nowszymi schematami podpisu, takimi jak EdDSA. [8]

DSA jest objęty amerykańskim patentem 5,231,668, złożonym (https://patents.google.com/patent/US5231668) 26 lipca 1991 roku, a obecnie wygasłym, i przypisywanym Davidowi W. Kravitzowi[9], byłemu pracownikowi NSA. Patent ten został przyznany "Stanom Zjednoczonym Ameryki reprezentowanym przez Sekretarza Handlu w Waszyngtonie", a NIST udostępnił ten patent na całym świecie bez opłat licencyjnych. [10] Claus P. Schnorr twierdzi, że jego amerykański patent 4,995,082 (https://patents.google.com/patent/US4995082) (również obecnie wygasły) obejmował DSA; Twierdzenie to jest kwestionowane. [11]

W 1993 roku Dave'owi Banisarowi udało się uzyskać potwierdzenie, za pośrednictwem FOIA, że algorytm DSA nie został zaprojektowany przez NIST, ale przez NSA. [12]

OpenSSH ogłosiło, że DSA ma zostać usunięte w 2025 roku. [13]

Operacja

Algorytm DSA obejmuje cztery operacje: generowanie klucza (który tworzy parę kluczy), dystrybucję klucza, podpisywanie i weryfikację podpisu.

1. Generowanie

Generowanie kluczy ma dwie fazy. Pierwsza faza to wybór *parametrów algorytmu*, które mogą być współdzielone przez różnych użytkowników systemu, podczas gdy druga faza oblicza pojedynczą parę kluczy dla jednego użytkownika.

Generowanie parametrów

- Wybieranie zatwierdzonej kryptograficznej funkcji skrótu H z długością wyjściową $|H|$ Bitów. W oryginalnym DSS H zawsze był SHA-1, ale silniejsze funkcje skrótu SHA-2 są zatwierdzone do użytku w obecnym DSS. [3][14] Jeśli $|H|$ jest większa niż długość modułu N, tylko skrajny lewy N bity wyjścia skrótu są używane.
- Wybieranie długości klucza L. Oryginalny ograniczony DSS L być wielokrotnością 64 między 512 a 1024 włącznie. NIST 800-57 zaleca długość 2048 (lub 3072) dla kluczy z okresami istnienia zabezpieczeń wykraczającymi poza rok 2010 (lub 2030). [15] Zob.
- Wybierz długość modułu N taki, że $N < L$ i $N \leq |H|$. FIPS 186-4 określa L i N , aby mieć jedną z wartości: (1024, 160), (2048, 224), (2048, 256) lub (3072, 256). [3]
- Wybierz ikonę N-liczba pierwsza bitu q.
- Wybierz ikonę L-liczba pierwsza bitu p taki, że $p - 1$ jest wielokrotnością q.
- Wybieranie liczby całkowitej h losowo z $\{2 \dots p - 2\}$.
- Liczyć $g := h^{(p-1)/q} \mod p$. W rzadkich przypadkach, gdy $g = 1$ Spróbuj ponownie z innym h. Powszechnie $h = 2$ jest używany. To modularne potęgowanie można efektywnie obliczyć, nawet jeśli wartości są duże.

Parametry algorytmu to (p, q, g). Mogą one być współdzielone przez różnych użytkowników systemu.

Klucze dla poszczególnych użytkowników

Biorąc pod uwagę zestaw parametrów, druga faza oblicza parę kluczy dla pojedynczego użytkownika:

- Wybieranie liczby całkowitej x losowo z $\{1 \dots q-1\}$.
- Liczyć $y := g^x \mod p$.

x to klucz prywatny, a y jest kluczem publicznym.

2. Dystrybucja kluczy

Sygnatariusz powinien opublikować klucz publiczny y. Oznacza to, że powinni wysłać klucz do odbiorcy za pomocą niezawodnego, ale niekoniecznie tajnego mechanizmu. Sygnatariusz powinien zachować klucz prywatny x tajny.

3. Podpisywanie

Wiadomość m jest podpisana w następujący sposób:

- Wybieranie liczby całkowitej k losowo z $\{1 \dots q-1\}$
- Liczyć $r := \left(g^k \mod p\right) \mod q$. W mało prawdopodobnym przypadku, gdy $r = 0$, zacznij od nowa z innym losowym k.
- Liczyć $s := \left(k^{-1} \left(H(m) + xr\right)\right) \mod q$. W mało prawdopodobnym przypadku, gdy $s = 0$, zacznij od nowa z innym losowym k.

Podpis jest (r, s)

Obliczanie k i r sprowadza się do utworzenia nowego klucza dla komunikatu. Modułowe potęgowanie w informatyce r jest najdroższą obliczeniowo częścią operacji podpisywania, ale może zostać obliczona, zanim komunikat będzie znany. Obliczanie odwrotności modularnej $k^{-1} \mod q$ jest drugą najdroższą częścią i może być również obliczona przed poznaniem wiadomości. Można go obliczyć za pomocą rozszerzonego algorytmu euklidesowego lub za pomocą małego twierdzenia Fermata jako $k^{q-2} \mod q$.

4. Weryfikacja podpisu

Można zweryfikować, czy podpis (r, s) jest prawidłowym podpisem wiadomości m Następujący sposób:

- Sprawdź, czy $0 < r < q$ i $0 < s < q$.
- Liczyć $w := s^{-1} \mod q$.
- Liczyć $u_1 := H(m) \cdot w \mod q$.
- Liczyć $u_2 := r \cdot w \mod q$.
- Liczyć $v := \left(g^{u_1} y^{u_2} \mod p\right) \mod q$.
- Podpis jest ważny wtedy i tylko wtedy, gdy $v = r$.

Poprawność algorytmu

Schemat podpisu jest poprawny w tym sensie, że weryfikator zawsze będzie akceptował autentyczne podpisy. Można to przedstawić w następujący sposób:

Po pierwsze, ponieważ $g = h^{(p-1)/q} \bmod p$, wynika z tego, że $g^q \equiv h^{p-1} \equiv 1 \mod p$ przez małe twierdzenie Fermata. Od $g > 0$ i q jest liczbą pierwszą, g musi mieć porządek q.

Sygnatariusz oblicza

$$s = k^{-1}(H(m) + xr) \bmod q$$

Więc

$$\begin{aligned} k &\equiv H(m)s^{-1} + xrs^{-1} \\ &\equiv H(m)w + xrw \pmod q \end{aligned}$$

Od g ma porządek q Mamy

$$\begin{aligned} g^k &\equiv g^{H(m)w} g^{xrw} \\ &\equiv g^{H(m)w} y^{rw} \\ &\equiv g^{u_1} y^{u_2} \pmod p \end{aligned}$$

Wreszcie prawidłowość DSA wynika z

$$\begin{aligned} r &= (g^k \bmod p) \bmod q \\ &= (g^{u_1} y^{u_2} \bmod p) \bmod q \\ &= v \end{aligned}$$

Czułość

W przypadku DSA entropia, tajność i unikatowość losowej wartości podpisu k są krytyczne. Jest to tak ważne, że naruszenie któregokolwiek z tych trzech wymagań może ujawnić atakującemu cały klucz prywatny. [16] Dwukrotne użycie tej samej wartości (nawet przy zachowaniu k secret), używając przewidywalnej wartości lub wycieku nawet kilku bitów k w każdym z kilku podpisów, wystarczy odsłonić klucz prywatny x. [17]

Problem ten dotyczy zarówno DSA, jak i algorytmu podpisu cyfrowego krzywej eliptycznej (ECDSA) – w grudniu 2010 r. grupa *failoverflow* ogłosiła odzyskanie klucza prywatnego ECDSA używanego przez Sony do podpisywania oprogramowania dla konsoli do gier PlayStation 3. Atak był możliwy, ponieważ Sony nie udało się wygenerować nowego losowego k dla każdego podpisu. [18]

Temu problemowi można zapobiec, wyprowadzając k deterministycznie z klucza prywatnego i skrótu wiadomości, zgodnie z opisem w RFC 6979 (https://datatracker.ietf.org/doc/html/rfc6979). Gwarantuje to, że k jest inny dla każdego $H(m)$ i nieprzewidywalne dla atakujących, którzy nie znają klucza prywatnego x.

Ponadto złośliwe implementacje DSA i ECDSA mogą być tworzone tam, gdzie k jest wybierany w celu podprogowego wycieku informacji za pomocą podpisów. Na przykład klucz prywatny offline może wycieknąć z doskonałego urządzenia offline, które udostępnia tylko niewinnie wyglądające podpisy. [19]

Implementacje

Poniżej znajduje się lista bibliotek kryptograficznych, które zapewniają obsługę DSA:

- Botan

- Dmuchany zamek
- cryptlib (biblioteka cryptlib)
- Krypto++
- libgcrypt powiedział:
- Pokrzywa
- OpenSSL (Protokół OpenSSL)
- wolfCrypt (wilcza krypta)
- Protokół GnuTLS

Zobacz też

- Arytmetyka modularna
- RSA (kryptosystem)
- ECDSA (Międzynarodowa Organizacja ds. T

Przypisy

1. Schneier, Bruce (1996). *Kryptografia stosowana* (https://archive.org/details/Applied_Cryptography_2nd_ed._B._Schneier). Wiley. ISBN 0-471-11709-9.
2. "FIPS PUB 186: Standard podpisu cyfrowego (DSS), 1994-05-19". (https://web.archive.org/web/20131213131144/http://www.itl.nist.gov/fipspubs/fip186.htm) *qcsrc.nist.gov*. [zarchiwizowane z tego adresu (http://www.itl.nist.gov/fipspubs/fip186.htm) (2013-12-13)].
3. "FIPS PUB 186-4: Standard podpisu cyfrowego (DSS), lipiec 2013 r (http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf)." (PDF). *csrc.nist.gov*.
4. "FIPS PUB 186-1: Standard podpisu cyfrowego (DSS), 1998-12-15" (https://web.archive.org/web/20131226115544/http://csrc.nist.gov/publications/fips/fips1861.pdf) (PDF). *csrc.nist.gov*. [zarchiwizowane z tego adresu (http://csrc.nist.gov/publications/fips/fips1861.pdf) (PDF)] (2013-12-26).
5. "FIPS PUB 186-2: Standard podpisu cyfrowego (DSS), 2000-01-27" (http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf) (PDF). *csrc.nist.gov*.
6. "FIPS PUB 186-3: Standard podpisu cyfrowego (DSS), czerwiec 2009 r (http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf)." (PDF). *csrc.nist.gov*.
7. "FIPS PUB 186-5: Standard podpisu cyfrowego (DSS), luty 2023 r (https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf)." (PDF). *csrc.nist.gov*.
8. "Standard podpisu cyfrowego (DSS)". (https://csrc.nist.gov/publications/detail/fips/186/5/draft) Departament Handlu Stanów Zjednoczonych. 31 października 2019 r. [dostęp 21 lipca 2020].
9. Dr David W. Kravitz (http://www.certicom.com/index.php/dr-david-kravitz) zarchiwizowane (https://web.archive.org/web/20130109092551/http://www.certicom.com/index.php/dr-david-kravitz) 9 stycznia 2013 r. w Wayback Machine
10. Wernera Kocha. "DSA i patenty" (https://lists.gnupg.org/pipermail/gnupg-devel/1997-December/014123.html)
11. "Sprawozdanie roczne CSSPAB za rok 1994". (https://web.archive.org/web/20090826042831/http://csrc.nist.gov/groups/SMA/ispab/documents/94-rpt.txt) 26 sierpnia 2009 r. [zarchiwizowane z tego adresu (http://csrc.nist.gov/groups/SMA/ispab/documents/94-rpt.txt) (26 sierpnia 2009)].
12. Neumann, Peter G. (2020-02-29). "The RISKS Digest Tom 14 Wydanie 59". (https://web.archive.org/web/20200229145033/https://catless.ncl.ac.uk/Risks/14/59) [zarchiwizowane z tego adresu (2020-02-29)]. [dostęp 2023-10-03].
13. "OpenSSH ogłasza harmonogram usuwania DSA [LWN.net]". (https://lwn.net/Articles/958048/) *lwn.net*. [dostęp 11 stycznia 2024].

14. "FIPS PUB 180-4: Secure Hash Standard (SHS), marzec 2012 r (http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf)." (PDF). *csrc.nist.gov*.
15. "Specjalna publikacja NIST 800-57" (https://web.archive.org/web/20140606050814/http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf) (PDF). *csrc.nist.gov*. [zarchiwizowane z tego adresu (http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf) (PDF)] (2014-06-06).
16. "Katastrofa Debiana PGP, która prawie była". (http://rdist.root.org/2009/05/17/the-debian-pgp-disaster-that-almost-was/) *Laboratoria korzeniowe RDIST*. 18 maja 2009 r.
17. DSA k-wartość Wymagania (https://rdist.root.org/2010/11/19/dsa-requirements-for-random-k-value/)
18. Bendel, Mike (2010-12-29). "Hakerzy opisują zabezpieczenia PS3 jako epicką porażkę, uzyskaj nieograniczony dostęp". (http://exophase.com/20540/hackers-describe-ps3-security-as-epic-fail-gain-unrestricted-access/) Exophase.com. [dostęp 2011-01-05].
19. Verbücheln, Stephan (2 stycznia 2015). "Jak doskonałe portfele offline mogą nadal wyciekać klucze prywatne Bitcoin". arXiv:1501.00447 (https://arxiv.org/abs/1501.00447) [cs. CR]. (https://arxiv.org/archive/cs.CR)

External links

- FIPS PUB 186-4: Digital Signature Standard (DSS) (http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf), the fourth (and current) revision of the official DSA specification.
- Recommendation for Key Management -- Part 1: general (https://web.archive.org/web/20140606050814/http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf), NIST Special Publication 800-57, p. 62–63
