ZESZYTY NAUKOWE AKADEMII MARYNARKI WOJENNEJ
ROK LIV NR 2 (193) 2013

Przemysław Rodwald
Akademia Marynarki Wojennej
Wydział Nawigacji i Uzbrojenia Okrętowego, Instytut Hydroakustyki
81-103 Gdynia, ul. J. Śmidowicza 69
e-mail: p.rodwald@amw.gdynia.pl

KRYPTOGRAFICZNE FUNKCJE SKRÓTU

STRESZCZENIE

Celem artykułu jest przegląd informacji dotyczących funkcji skrótu oraz przedstawienie najnowszych osiągnięć kryptografii w tym zakresie. Wyjaśnione są podstawowe pojęcia dotyczące funkcji skrótu, ich zastosowanie oraz metody ataków. Pokazany jest bieżący stan kryptoanalizy znanych i powszechnie stosowanych funkcji skrótu: MD4, MD5, SHA. Na zakończenie omówiona jest przyszłość funkcji skrótu i zakończony konkurs na nowy standard funkcji SHA-3.

Słowa kluczowe:
kryptografia, kryptoanaliza, funkcje skrótu.

WSTĘP

Funkcje skrótu (*hash functions*) odgrywają bardzo ważną rolę we współczesnej kryptografii. Wiele osób nawet nie zdaje sobie sprawy, jak często każdego dnia nieświadomie wykorzystuje funkcje skrótu, na przykład podczas korzystania z Internetu czy też podczas używania kart płatniczych. W 1990 roku Rivest przedstawił funkcje skrótu MD4 [11]. Stała się ona podstawą do tworzenia innych funkcji, takich jak MD5, SHA czy RIPEMD. Po okresie znikomego zainteresowania kryptoanalizą funkcji skrótu ostatnie lata przyniosły wiele ciekawych prac w tej dziedzinie. Większość powszechnie używanych funkcji z rodziny MD/SHA została skompromitowana. W środowisku kryptologów zaczęto zadawać sobie pytanie, co dalej.

DEFINICJA, WŁAŚCIWOŚCI

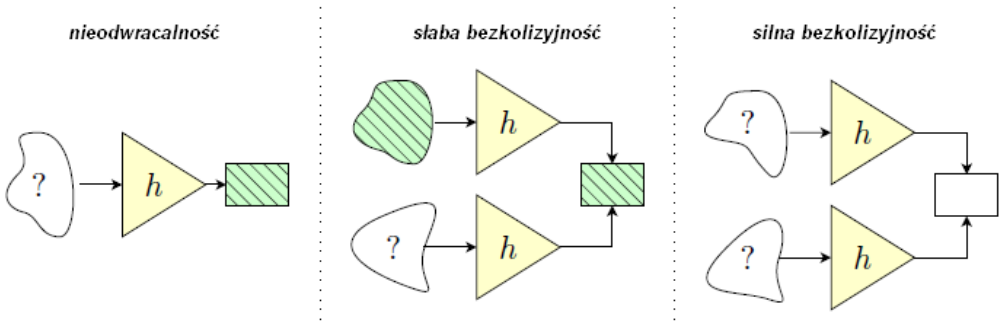
Pod pojęciem funkcji skrótu h rozumie się odwzorowanie wiadomości m o dowolnej, skończonej długości, w ciąg bitów o określonej, stałej długości n:

$$h : \{0,1\}^* \to \{0,1\}^n \text{, gdzie: } \{0,1\}^* = \bigcup_{i \in \mathrm{N}} \{0,1\}^i \text{, } \mathrm{N} = \{0, 1, 2, \ldots\}, \; n \in \mathrm{N}. \quad (1)$$

Podstawowym zadaniem funkcji skrótu jest ochrona integralności danych, czyli ochrona przed ich zmodyfikowaniem w sposób nieautoryzowany. Od funkcji skrótu oczekuje się, aby w sposób szybki i jednoznaczny identyfikowały dane cyfrowe. Oznacza to w praktyce, iż nawet małe zmiany (na przykład zmiana jednego bitu) w skracanym ciągu danych powinny dać w rezultacie zupełnie inny skrót (średnio zmianę połowy bitów skrótu). Nie powinno być także możliwości odtworzenia wiadomości oryginalnej, jeżeli ma się tylko jej skrót, oraz nie powinno być możliwości utworzenia dwóch różnych wiadomości dających ten sam skrót. Wymagania stawiane funkcjom skrótu można przedstawić jako następujące właściwości:

1. N i e o d w r a c a l n o ś ć (*preimage resistance*, *non-invertiblity*): dany jest skrót $h(m)$, wiadomość m jest nieznana. Znalezienie wiadomości m jest obliczeniowo trudne.
2. S ł a b a b e z k o l i z y j n o ś ć (*2nd preimage resistance*, *weak collision resistance*): dany jest skrót $h(m)$ i odpowiadająca mu wiadomość m. Znalezienie wiadomości $m' \neq m$, takiej że $h(m) = h(m')$, jest obliczeniowo trudne.
3. S i l n a b e z k o l i z y j n o ś ć (*collision resistance*, *strong collision resistance*): obliczeniowo trudne jest znalezienie dowolnej pary różnych wiadomości m' i m, takich że $h(m) = h(m')$.

Pod pojęciem problemu trudnego obliczeniowo rozumie się atak, którego złożoność obliczeniowa w praktyce jest tak duża (teoretycznie: wykładnicza), że przy współcześnie istniejącej technice oraz stanie wiedzy staje się on praktycznie niewykonalny w rozsądnym czasie. Powyższe, podstawowe właściwości w sposób graficzny zostały przedstawione na rysunku 1.



Rys. 1. Podstawowe właściwości funkcji skrótu

Źródło: opracowanie własne.

W literaturze szeroko stosowane są również następujące określenia funkcji skrótu:

— jednokierunkowa funkcja skrótu (OWHF — *One Way Hash Function*) — spełnienie własności nieodwracalności i słabej bezkolizyjności;
— bezkolizyjna funkcja skrótu (CRHF — *Collision Resistance Hash Function)* — spełnienie własności silnej bezkolizyjności.

ZASTOSOWANIA

Spośród algorytmów kryptograficznych to właśnie funkcje skrótu należą do najbardziej wszechstronnych. Stosuje się je przy weryfikacji integralności komunikatów (MDC — *Modification Detection Code*), w celu zapewnienia, że komunikat nie został w żaden sposób zmodyfikowany, czy też jako kody uwierzytelniające (MAC — *Message Authentication Code*), gwarantujące nie tylko niezmienność wiadomości, ale również zapewniające zweryfikowanie jej nadawcy — pod warunkiem sparametryzowania funkcji skrótu kluczem.

Innym ważnym zastosowaniem funkcji skrótu są schematy podpisu cyfrowego, które znacznie przyspieszają proces podpisywania wiadomości. Zamiast czasochłonnego szyfrowania i deszyfrowania wiadomości (niejednokrotnie bardzo dużych) wystarczy zaszyfrować sam skrót uzyskany przy użyciu funkcji skrótu (znacznie szybszych od algorytmów szyfrujących).

Pośród kolejnych zastosowań funkcji skrótu warto wymienić ich powszechność przy generowaniu i przechowywaniu haseł, gdyż we współczesnych systemach informatycznych hasła nie są przechowywane w postaci jawnej, lecz właśnie w postaci generowanych przez nie skrótów. Używa się ich także na szeroką skalę w celu badania integralności programów, różnego rodzaju łat i uaktualnień czy też sygnatur wirusów. Funkcje skrótu znalazły też szerokie zastosowanie w różnych protokołach, między innymi SSL, SSH, IPsec. Stosowane są również przy generowaniu ciągów pseudolosowych.

BUDOWA FUNKCJI SKRÓTU

Od czasu, gdy w 1989 roku Merkle i Damgård niezależnie przedstawili iteracyjną strukturę konstrukcji funkcji skrótu (rys. 2.), stała się ona najbardziej popularnym i powszechnie stosowanym podejściem projektowym. Polega ono na konstruowaniu funkcji skrótu danej wzorem (1) za pomocą funkcji kompresującej φ o określonej długości wektora wejściowego

$$\varphi : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n , \tag{2}$$

gdzie:

t — długość bloku funkcji odpowiadająca długości bloku skracanej wiadomości;

n — długość zmiennej łańcuchowej odpowiadającej długości wektora inicjującego i najczęściej równej długości skrótu.

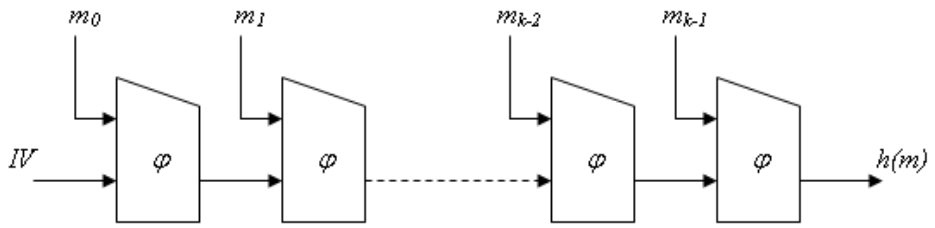
Dla konstrukcji tej prawdziwe jest stwierdzenie, iż jeśli funkcja kompresująca φ jest odporna na kolizje, to wynikowa funkcja skrótu h jest także odporna na kolizje.

Algorytm wyznaczania skrótu dla wiadomości m podzielonej na k bloków równej długości przedstawia się następująco:

$$\begin{aligned} &CV_0 = IV \\ &CV_{i+1} = \varphi(CV_i, m_i), \quad i = 0,1,...,k-1, \\ &h(m) = CV_k \end{aligned} \tag{3}$$

gdzie

IV — wektor inicjujący ustalony *a priori* dla danej funkcji.



Rys. 2. Schemat struktury Merklego-Damgårda

Źródło: opracowanie własne.

Niestety, ostatnie ataki pokazują słabość konstrukcji Merklego-Damgårda (MD) i jej podatność na atak wieloblokowy. We współcześnie projektowanych funkcjach skrótu (na przykład pośród kandydatów na nowy standard SHA-3) klasyczny schemat iteracyjny MD jest stosowany sporadycznie. Raczej stosuje się jego zmodyfikowane wersje [2], [11] lub proponuje zupełnie nowe podejścia (np. *sponge* [1]).

ATAKI NA FUNKCJE SKRÓTU

Wśród metod łamania funkcji skrótu należy wyróżnić klasę ataków niewykorzystujących słabości wewnętrznej struktury analizowanej funkcji skrótu (brutalny, urodzinowy, wieloblokowy) oraz ataki znajdujące słabości przekształceń zastosowanych wewnątrz funkcji (różnicowy, liniowy).

Atak brutalny

Istotą ataku brutalnego jest znalezienie dowolnej wiadomości $m' \neq m$, która po skróceniu da zadany skrót $h(m') = h(m)$. Atak ten polega na przeszukiwaniu losowego zbioru wiadomości i porównywaniu z zadaną wartością skrótu. Jest najwolniejszym z ataków, a jego złożoność, zarówno obliczeniowa, jak i pamięciowa, wynosi $O(2^n)$.

Atak urodzinowy

Istotą ataku urodzinowego jest znalezienie dowolnej pary wiadomości m i m', takich że $h(m') = h(m)$. Atak ten oparty jest na paradoksie dnia urodzin, który głosi,

że prawdopodobieństwo tego, iż spośród dwudziestu trzech osób dwie mają urodziny w tym samym dniu wynosi więcej niż jedna druga. Atak ten polega na wygenerowaniu i zapamiętaniu $2^{n/2}$ wiadomości i tego rzędu jest złożoność ataku urodzinowego.

Atak wieloblokowy

Istotą tego ataku jest znalezienie dowolnej pary złożonych wiadomości $m_1||m_2||..||m_k$ i $m'_1||m'_2||..||m'_k$, takich że $h(m'_1||m'_2||..||m'_k) = h(m_1||m_2||..||m_k)$. Atak ten stosowany jest przeciwko strukturze MD. W ataku tym jako wyniki cząstkowe wykorzystuje się zarówno pseudokolizje (*pseudo-collision*), jak i prawiekolizje (*near-collision*).

Atak różnicowy

Istotą tego ataku jest znalezienie dwóch wiadomości dających ten sam skrót, z wykorzystaniem niedoskonałości zastosowanej funkcji kompresującej. Różnica jest zazwyczaj definiowana jako funkcja logiczna *xor*, a filozofia ataku wykorzystuje fakt, iż poprzez zmianę kilku bitów w wiadomości prawdopodobne jest zniwelowanie różnicy wewnątrz funkcji kompresującej po kilku rundach.

BIEŻĄCY STAN KRYPTOANALIZY

Obecny stan kryptoanalizy znanych i powszechnie używanych funkcji skrótu przedstawia się następująco.

MD4

Funkcja skrótu MD4, która dała początek całej rodzinie MD/SHA, została stworzona w 1990 roku przez Rona Rivesta [11]. Rok 1992 przynosi atak na dwie ostatnie rundy algorytmu autorstwa pary: den Boer i Bosselaers. Hans Dobbertin w 1997 roku najpierw wykazuje, iż dwie pierwsze rundy algorytmu nie są jednokierunkowe, a następnie przedstawia algorytm znajdowania kolizji z prawdopodobieństwem

2^{-22}. Rok 2004 to kres funkcji MD4. Chińczycy pod przewodnictwem profesor Wang łamią ją, przedstawiając wzór na generowanie kolizji [14].

MD5

Funkcja MD5 autorstwa Rivesta jest bezpośrednią następczynią MD4. Została zaprezentowana w 1991 roku po sugestiach Dobbertina odnoście słabości poprzedniczki. Jednak już w 1993 roku Boer i Bosselaers znajdują pseudokolizje dla tej funkcji, a trzy lata później Dobbertin znajduje kolizje dla funkcji kompresującej algorytmu MD5. Rok 2004 to także obnażenie słabości tej funkcji. Wang przedstawia przepis na znajdowanie kolizji dla dwublokowych wiadomości. Znalezienie kolizji na komputerze klasy PC zajmuje około godziny. W marcu 2006 roku Vlastimil Klima publikuje algorytm [6] znajdujący kolizje w czasie do jednej minuty, wykorzystując metodę zwaną tunelowaniem.

RIPEMD

Funkcja skrótu RIPEMD powstała w ramach projektu Unii Europejskiej o nazwie RIPE (RACE Integrity Primitives Evaluation) realizowanego w latach 1988–1992. Kilka lat później powstaje wzmocniona wersja algorytmu o nazwie RIPEMD-160, której projektantami są Dobbertin, Bosselaers oraz Preneel [4]. Już w 1995 roku sam Dobbertin udowadnia, iż dwie rundy funkcji kompresującej RIPEMD nie są odporne na kolizje. Następnie Wang łamie ją „na kartce papieru", przedstawiając wzór na znajdowanie kolizji [14]. Funkcja RIPEMD-160 do dnia dzisiejszego nie poddała się atakom kryptoanalityków.

SHA-0

Początek rodziny funkcji SHA (Secure Hash Algorithm) datuje się na 1993 rok. Wówczas NSA (National Security Agency) poprzez NIST (National Institute of Standards and Technology) publikuje pierwszą funkcję z tej rodziny, nazywaną często SHA-0. Historia kryptoanalizy przedstawia się następująco:

1) Chabaud i Joux w 1998 roku przedstawiają atak znajdujący kolizje o złożoności 2^{61}.

2) Biham i Chen w 2004 roku znajdują prawiekolizje (18 bitów różnicy) oraz pełną kolizję dla zredukowanej do 62 rund (z 80) wersji algorytmu.
3) 12 sierpnia 2004 roku Joux, Carribault, Lemuet i Jalby przedstawiają algorytm znajdujący kolizję o złożoności 2^{51}.
4) 17 sierpnia 2004 roku na konferencji Crypto '04 Chińczycy anonsują atak o złożoności 2^{40}, którego szczegółów jednak nie podali.
5) X. Wang, Yin i Yu [15] w lutym 2005 roku przedstawiają atak o złożoności 2^{39}.
6) Pełna kolizja została przedstawiona wcześniej przez Jouxa [5].

SHA-1

Dwa lata później opublikowany zostaje algorytm SHA-1, który zastępuje swojego poprzednika ze względu na nieujawnione oficjalnie wady. Następczyni algorytmu, czyli funkcja SHA-1, także okazała się podatna na ataki:

1) W 2004 roku następują dwa niezależne ataki na zredukowaną do 53 rund (z 80) wersję algorytmu przeprowadzone przez pary: Biham, Chen oraz Rijmen, Oswald.
2) W lutym 2005 roku Chińczycy przedstawiają atak o złożoności 2^{69} [16], po czym w sierpniu ulepszają go do złożoności 2^{63} [17].

SHA-2

W 2001 roku NIST publikuje ulepszoną wersję funkcji SHA, dając jej roboczą nazwę SHA-2 i czekając na komentarze. W jej skład wchodzą trzy funkcje: SHA-256, SHA-384 i SHA-512. Rok później rodzina ta staje się standardem opublikowanym jako FIPS PUB 180-2. Funkcje z rodziny SHA-2 na dzień dzisiejszy okazały się odporne na znane ataki.

NOWY STANDARD

Funkcje, które uważano za bezpieczne i które były powszechnie stosowane (np. MD5, SHA-0), okazały się podatne na współczesne ataki. Między innymi z tego powodu pod koniec 2007 roku NIST ogłosił konkurs na nową funkcję skrótu pod

nazwą SHA-3 [8]. Rok 2008 przyniósł środowisku kryptoanalityków 64 propozycje nowych funkcji skrótu — propozycje na nowy standard SHA-3. Część z nich została bardzo szybko złamana i nie dostała się nawet do pierwszej rundy konkursu. Kolejne rundy konkursu eliminowały kolejnych kandydatów: w rundzie pierwszej było 51 funkcji, w drugiej 14, a do finałowej trzeciej rundy zostało zakwalifikowanych 5 funkcji (BLAKE, Grøstl, JH, Keccak, Skein). Analizując raport wydany przez NIST [10], można zobaczyć, iż wybrane funkcje cechują się dużą różnorodnością ich struktury wewnętrznej:

- BLAKE — zbudowany jest w oparciu o uproszczony schemat iteracyjny HAIFA [2], opiera się na strukturze niezrównoważonej sieci Feistela, a jego funkcja kompresująca opiera się na trzech podstawowych operacjach bitowych (+, <<<, $\oplus$) — tak zwana konstrukcja ARX (od słów Addition, Rotation, XOR).
- Grøstl — zbudowany jest na strukturze MD z dodatkowym przekształceniem wyjściowym, koniecznym, gdyż wektor stanu jest dwukrotnie dłuższy niż generowany skrót; funkcja kompresująca oparta jest natomiast na skrzynkach podstawieniowych (*substitution box*, *S-box*), takich samych jak w algorytmie AES.
- JH — opiera się na zrównoważonej strukturze Feistela z funkcją kompresującą opartą na skrzynkach podstawieniowych o rozmiarze 4 x 4.
- Keccak — wykorzystuje zupełnie nowe, obiecujące podejście projektowe, a mianowicie konstrukcję gąbki (*sponge construction*).
- Skein — opiera się na zrównoważonej strukturze Feistela z funkcją kompresującą wykorzystującą konstrukcję ARX.

W 2012 roku NIST ogłosił [9] zwycięzcę konkursu; została nim funkcja Keccak. W końcowym raporcie z konkursu przedstawione zostały następujące argumenty uzasadniające wybór zwycięzcy: elegancka konstrukcja, duży margines bezpieczeństwa, doskonała wydajność w implementacjach sprzętowych. Dodatkowo podkreślony został fakt zastosowania zupełnie nowego podejścia projektowego, całkowicie rożnego od rodziny MD/SHA. Z jednej strony mało prawdopodobne wydaje się, iż jakiś atak skutecznie zagrozi obu funkcjom. Z drugiej strony projektanci systemów bezpieczeństwa mają teraz wybór pomiędzy dwoma różnymi algorytmami i mogą używać tego, który lepiej spełnia ich wymagania.

W 2013 roku ma powstać nowy standard funkcji skrótu FIPS (*Federal Information Processing Standard*).

PODSUMOWANIE

Ogromny wzrost zainteresowania kryptoanalizą funkcji skrótu w ostatnich latach oraz wynikające z tego ataki (głównie Chińczyków) stawiają pod znakiem zapytania bezpieczeństwo stosowania najpopularniejszych funkcji skrótu. Aktualnie najszerzej używane w zastosowaniach komercyjnych funkcje MD5 i SHA-1 zostały skompromitowane. Stojąc przed problemem konstruowania systemów zawierających funkcje skrótu, należy zatem wybierać: ciągle uważane za bezpieczne te z rodziny SHA-2 lub zaufać nowo tworzonemu standardowi SHA-3.

BIBLIOGRAFIA

[1] Bertoni G., Daemen J., Peeters M., Van Assche V., *Sponge Functions*, ECRYPT Hash Workshop, Barcelona 2007.

[2] Biham E., Dunkelman O., *A Framework for Iterative Hash Functions — HAIFA*, The Second Cryptographic Hash Workshop, Santa Barbara 2006.

[3] Damgård I., *A design principle for hash functions*, Advances in Cryptology — CRYPTO 1989, LNCS 435, Springer-Verlag, 1989.

[4] Dobbertin H., Bosselaers A., Preneel B., *RIPMEMD-160: A Strengthened Version of RIPMMD*, Advances in Cryptology, FSE '96, LNCS 1039, Springer-Verlag, 1996.

[5] Joux A., *Collisions in SHA-0*, CRYPTO 2004, Rump Session, 2004.

[6] Klima V., *Tunnels in Hash Functions: MD5 Collisions Within a Minute*, Cryptology ePrint Archive, http://eprint.iacr.org/2006/105, 2006.

[7] Merkle R., *A Fast Software One-Way Hash Function*, 'Journal of Cryptology', 1990, Vol. 3, No 1.

[8] NIST, *Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family*,

http://csrc.nist.gov/groups/ST/ hash/documents/FR_Notice_Nov07.pdf, 2007.

[9] NIST, *SHA-3 Selection Announcement*,

http://csrc.nist.gov/groups/ST/hash/sha-3/sha-3_selection_announcement.pdf, 2012.

[10] NIST, *Status Report on the Second Round of the SHA-3 Cryptographic Hash Algorithm Competition,* http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/ documents/Round2_Report_NISTIR_7764.pdf, 2011.

[11] Rivest R., *The MD4 message-digest algorithm*, Advances in Cryptology, Proc. Crypto '90, LNCS 597, Springer-Verlag, 1991.

[12] Rodwald P., Stokłosa J., *Family of Parameterized Hash Algorithms*, International Conference on Emerging Security Information, Systems and Technologies, IEEE Computer Society Order Number E3329, Francja, 2008.

[13] Schneier B., *Nist hash workshop liveblogging*, Schneier on Security Internet Blog, http://www.schneier.com/blog/archives/2005/11/, 2005.

[14] Wang X., Feng D., Lai X., Yu H., *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, Cryptology ePrint Archive, http://eprint.iacr.org/ 2004/199, 2004.

[15] Wang X., Yu H., Yin Y., *Efficient Collision Search Attacks on SHA-0*, Advances in Cryptology, Crypto '05, LNCS 3621, Springer-Verlag, 2005.

[16] Wang X., Yin A. L., Yu H., *Finding Collisions in the Full SHA-1*, Advances in Cryptology, Crypto '05, LNCS 3621, Springer-Verlag, 2005.

[17] Wang X., Yao A., Yao F., *New Collision search for SHA-1*, Rump Session Crypto '05, 2005.

CRYPTOGRAPHIC HASH FUNCTIONS

ABSTRACT

The article presents a synthesis of information about the hash function and shows the latest developments in this field of cryptography. Basic concepts of the hash function are explained:

definition, properties, classification, usage of the hash function and methods of attacks. The current state of cryptanalysis of known and commonly used hash functions (MD, SHA) is shown as well as consequences coming from this. At the end the attention will be paid to the future of the hash function and the current state of art in the competition for developing the new standard of SHA-3 function.

Keywords:
cryptography, cryptanalysis, hash functions.