W poszukiwaniu funkcji skrótu

Stanisław P. RADZISZOWSKI[*]
Michał ADAMASZEK

Zacznijmy od wyjaśnienia tytułu. *Funkcja skrótu* (nazwijmy ją H) służy do przyporządkowywania dowolnie długim wiadomościom krótkich liczb („odcisków palca"). Wiadomością może być plik, obrazek, film w postaci cyfrowej lub inna, zupełnie dowolna porcja danych (x), którą będziemy traktować po prostu jako ciąg bitów. Kodowany ciąg x może być bardzo długi, za to wynik $H(x)$ powinien być krótki, na przykład kilkusetbitowy.

Do czego przydają się takie funkcje? Mogą służyć do ochrony przed modyfikacjami danych, a na ich popularność wpływa rosnąca dostępność Internetu, a wraz z nim możliwość wymieniania się przeróżnymi plikami. Wiąże się to z różnymi zagrożeniami: skąd mamy wiedzieć, że pobieramy kopię identyczną z oryginalną, a nie taką, która została celowo zmodyfikowana (np. poprzez wprowadzenie wirusa)? Dobra funkcja skrótu powinna służyć między innymi do wykrywania takich machlojek: jeżeli autor pliku x opublikuje $H(x)$, to ktoś, kto chce stwierdzić, czy ma wierną kopię x, może sam szybko obliczyć $H(x)$ i porównać ją z opublikowaną wartością.

Zamiast funkcja skrótu mówi się nieraz funkcja haszująca, choć to drugie pojęcie jest mniej restrykcyjne, bo funkcja haszująca musi po prostu dobrze rozpraszać dane i nie pełni funkcji kryptograficznej.

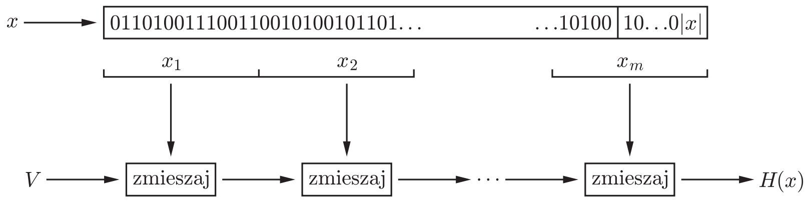
Można pomyśleć, że dobrą sygnaturą pliku jest, na przykład, jego rozmiar. Niestety, istnieją rozmaite sposoby podrabiania plików w taki sposób, aby ich rozmiar nie zmieniał się. Podobną wadę mają inne zbyt proste techniki. Dobra funkcja skrótu H powinna spełniać następujące warunki:

- (1) szybko obliczalna on-line, w niewielkiej pamięci,
- (2) bardzo trudno odwracalna: dla danego z znalezienie x takiego, że $H(x) = z$, powinno być bardzo kosztowne obliczeniowo,
- (3) odporna na kolizje: znalezienie dwóch wiadomości x i y, dla których $H(x) = H(y)$, także powinno być bardzo trudne.

Zatem bardzo łatwo obliczyć wartość skrótu, ale niesłychanie trudno wykonać operację odwrotną. Warunek (3) oznacza, że funkcja H jest „praktycznie różnowartościowa" – w zasadzie nie mamy szans trafić na dwa ciągi bitów o tym samym skrócie. Jak widać, funkcja H powinna dobrze „mieszać" wszystkie bity wejściowego ciągu. Funkcje haszujące mają wiele zastosowań: do obliczania sum kontrolnych, uwierzytelniania, przechowywania haseł czy w technologii podpisów cyfrowych.

Większość używanych funkcji skrótu stosuje tzw. schemat Merkle-Damagarda. Polega on na tym, że ciąg wejściowy x jest dzielony na bloki ustalonej długości b: $x = x_1x_2\dots x_m$, $|x_i| = b$. Ostatni blok dopełniamy zerami i binarnym zapisem długości tekstu x na ustalonej liczbie bitów, tak aby również osiągnąć długość b. Ponadto mamy określony pewien inicjalny ciąg bitów V, po czym wykonujemy pętlę (rys.1):

$H_0 := V;$
for $k := 1$ **to** m **do**
  $H_k = \mathit{zmieszaj}(H_{k-1}, x_k);$
return $H_m;$



Rys.1

Cała sztuka polega na doborze odpowiedniej funkcji kompresującej *zmieszaj*, której zadaniem jest zmieszanie zawartości i-tego bloku ze skrótem poprzednich bloków. Podkreślmy, że nie ma na to żadnego uniwersalnego sposobu – wszystko opiera się na kreatywności twórców pomysłu i intuicji dotyczącej tego, jaki sposób mieszania będzie trudno odwracalny. W praktyce popularne są dwa

[*] Rochester Institute of Technology