

Ochrona i poufność danych

Maciej Nikodem

03.11.2011

- Funkcja skrótu (ang. *Hash function*) – przyporządkowuje dowolnie dużemu argumentowi, krótką, zwykle posiadającą stały rozmiar niezależny od argumentu, wartość.
- Jednokierunkowa funkcja skrót (ang. *One-Way Hash Function*) – na podstawie wyniku funkcji nie da się określić argumentu,
- Kryptograficzna funkcja skrót (ang. *Collision Resistant Hash Function*) – funkcja jednokierunkowa z własnościami odporności na kolizje,
- Funkcje skrót z kluczem (ang. *Message Authentication Code*).

$$h : \{0, 1\}^* \rightarrow \{0, 1\}^n$$

- $\|D\| \gg \|R\|$ - kompresja,
- jeśli h jest losowa i przeciwdziedzina h jest zbiór wszystkich liczb n -bitowych, to

$$\mathcal{P}(h(d_1) = h(d_2)) \approx \frac{1}{2^n}$$

- łatwość obliczania,
- efekt lawionowy – niewielka zmiana argumentów, generuje diametralnie różne wartości funkcji skrótu,

Dodatkowo, kryptograficzne funkcje skrótu są:

- jednokierunkowe,
- odporne na kolizje,

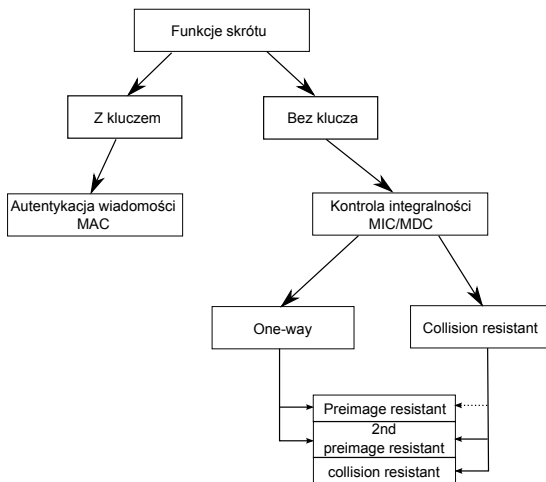
Jednokierunkowość:

- preimage resistance – mając dane $h(x)$ jest trudnym odszukanie x ,
- 2nd preimage resistance – mając ustalone x_1 jest trudnym odszukanie x_2 takiego, że $h(x_1) = h(x_2)$,

Odporność na kolizje:

- collision resistance – jest trudne odszukanie dwóch $x_1 \neq x_2$ takich, że $h(x_1) = h(x_2)$.

Klasyfikacja kryptograficznych funkcji skrótu



- schematy podpisów cyfrowych,
- zapewnianie integralności danych – Message Integrity Code, Modification Detection Code,
- autentykacja wiadomości – Message Authentication Code,
- protokoły uwieczniania.

Wymagania w zastosowaniach funkcji skrótu

Zastosowanie	Własności		
	preimage resistant	2nd preimage resistant	collision resistant
MIC - hasła	+		
MIC - autentykacja		+	+
MIC - podpisy cyfrowe	+	+	+
MAC (klucz nieznanemu atakującemu)	+	+	+
MAC (klucz znany atakującemu)		+	

- hasła,
- protokoły wyzwanie–odpowiedź,
- protokoły z wiedzą zerową,

- wzajemne zidentyfikowanie i/lub potwierdzenie tożsamości komunikujących się stron,
- zagwarantowanie, że po identyfikacji/autoryzacji Alicji i Boba, Bob nie będzie mógł podszyć się pod Alicję (i odwrotnie) w komunikacji z Ewą,
- zminimalizowanie prawdopodobieństwa skutecznego podszycia się Ewy pod Alicję i poprawnego przeprowadzenia identyfikacji/autentykacji z Bobem.

- coś wiedzieć – hasła, PINy, klucze,
- coś mieć – karta μ P, token, paszport,
- być kimś – cechy biometryczne,

- wzajemność,
- efektywność obliczeń,
- efektywność komunikacji,
- on-line,

- "szyfrowane" pliki haseł,
- reguły haseł,
- spowalnianie wprowadzania haseł,
- rozszerzanie haseł,
- passphrases.

- minimalna liczba znaków hasła,
- rodzaj znaków w hasle,
- częstotliwość zmiany haseł,

Liczba znaków hasła	Liczność zbioru dostępnych znaków			
	26	36	62	95
5	23.5	25.9	29.8	32.9
6	28.2	31.0	35.7	39.4
8	37.6	41.4	47.6	52.6
10	47.0	51.7	59.5	65.7

Tabela: Logarytm o podstawie 2 z liczby haseł dla danego zestawu znaków i rozmiaru haseł.

- powtórzenie hasła lub skrótu hasła,
- przegląd zupełny

Liczba znaków hasła	Liczność zbioru dostępnych znaków			
	26	36	62	95
5	0.67h	3.4h	51h	430h
6	17h	120h	130 dni	4.7 lat
8	1.3 lat	18 lat	1400 lat	42 000 lat
10	890 lat	23 000 lat	$5.3 \cdot 10^6$ lat	$3.8 \cdot 10^8$ lat

Tabela: Czasy przeglądu zupełnego dla danego zestawu znaków i rozmiaru haseł.

- atak słownikowy, odgadywanie haseł

- spowalnianie wprowadzania haseł – wydłużanie czasu pomiędzy kolejnymi wprowadzeniami hasła,
- rozszerzanie haseł – powiększanie przestrzeni klucza,
- passphrases

Light Amplification by Stimulated Emission of Radiation.



LASER

Generacja haseł

$$w_0 \rightarrow w_1 \rightarrow w_2 \rightarrow \dots \rightarrow w_t$$

Serwer zapamiętuje hasło w_t ,

Użytkownik dostaje listę haseł, bez w_t .

Identyfikacja

- użytkownik bierze ostatnie, nieużywane hasło – w_{t-1} ,
- hasło jest przesyłane do serwera,
- serwer wylicza $h(w_{t-1})$ i sprawdza czy $h(w_{t-1}) = w_t$,
- jeśli hasło jest poprawne to serwer zapamiętuje w_{t-1} ,

Nonce – liczba, która jest używana jednokrotnie:

- **Random number** – liczba losowa; ma zagwarantować "świeżość" wiadomości,
- **Sequence number** – jednoznaczny, kolejny, niepowtarzalny numer wiadomości,
- **Timestamp** – znaczniki czasu wiadomości; pozwalają wykrywać opóźnienia i gwarantują kolejność wykonania operacji,

Nonce – liczba, która jest używana jednokrotnie:

- **Random number** – liczba losowa; ma zagwarantować "świeżość" wiadomości,
- **Sequence number** – jednoznaczny, kolejny, niepowtarzalny numer wiadomości,
- **Timestamp** – znaczniki czasu wiadomości; pozwalają wykrywać opóźnienia i gwarantują kolejność wykonania operacji,

Wady poszczególnych rozwiązań:

- Random number – dodatkowa komunikacja, duża przestrzeń liczb,
- Sequence number – konieczność pamiętania ostatnio użytych liczb,
- Timestamp – konieczność synchronizacji zegarów,

Protokół wyzwanie odpowiedź z algorytmem symetrycznym

Z wykorzystaniem znacznika czasu

$$A \rightarrow B : E_K(t_A, B)$$

Z wykorzystaniem znacznika czasu

$$A \rightarrow B : E_K(t_A, B)$$

Z wykorzystaniem liczby losowej 1

$$A \leftarrow B : r_B$$

$$A \rightarrow B : E_K(r_B, B)$$

Z wykorzystaniem znacznika czasu

$$A \rightarrow B : E_K(t_A, B)$$

Z wykorzystaniem liczby losowej 1

$$A \leftarrow B : r_B$$

$$A \rightarrow B : E_K(r_B, B)$$

Z wykorzystaniem liczby losowej 2

$$A \leftarrow B : r_B$$

$$A \rightarrow B : E_K(r_A, r_B, B)$$

$$A \leftarrow B : E_K(r_B, r_A)$$

Protokół wyzwanie odpowiedź z funkcją skrótu

Z wykorzystaniem liczby losowej i alg.symetrycznego

$$A \leftarrow B : r_B$$

$$A \rightarrow B : E_K(r_A, r_B, B)$$

$$A \leftarrow B : E_K(r_B, r_A)$$

Protokół wyzwanie odpowiedź z funkcją skrótu

Z wykorzystaniem liczby losowej i alg.symetrycznego

$$\begin{aligned}A \leftarrow B &: r_B \\A \rightarrow B &: E_K(r_A, r_B, B) \\A \leftarrow B &: E_K(r_B, r_A)\end{aligned}$$

Z wykorzystaniem liczby losowej i funkcji skrótu z kluczem

$$\begin{aligned}A \leftarrow B &: r_B \\A \rightarrow B &: r_A, h_K(r_A, r_B, B) \\A \leftarrow B &: h_K(r_B, r_A, A)\end{aligned}$$

Celem stosowania kryptografii asymetrycznej jest udowodnienie, że znamy klucz prywatny. Może być to zrealizowane poprzez:

- deszyfrowanie wyzwania zaszyfrowanego kluczem publicznym,
- podpisanie cyfrowe otrzymanego wyzwania.

Celem stosowania kryptografii asymetrycznej jest udowodnienie, że znamy klucz prywatny. Może być to zrealizowane poprzez:

- deszyfrowanie wyzwania zaszyfrowanego kluczem publicznym,
- podpisanie cyfrowe otrzymanego wyzwania.

$$A \leftarrow B : r_B$$

$$A \rightarrow B : cert_A, r_A, B, Sig_A(r_A, r_B, B)$$

$$A \leftarrow B : cert_B, A, Sig_B(r_B, r_A, A)$$

- hasła – weryfikujący (ang. *verifier*) otrzymuje informację, która pozwala mu podszyć się pod stronę potwierdzającą swoją tożsamość (ang. *claimant*),
- protokoły wyzwanie–odpowieź:
 - prezentowana dana ma ograniczoną ważność,
 - przesyłane wiadomości nie dają bezpośredniej wiedzy o tajnych danych,
 - nieuczciwy weryfikujący może umyślnie dobierać wyzwania.

- hasła – weryfikujący (ang. *verifier*) otrzymuje informację, która pozwala mu podszyć się pod stronę potwierdzającą swoją tożsamość (ang. *claimant*),
- protokoły wyzwanie–odpowieź:
 - prezentowana dana ma ograniczoną ważność,
 - przesyłane wiadomości nie dają bezpośredniej wiedzy o tajnych danych,
 - nieuczciwy weryfikujący może umyślnie dobrać wyzwania.

Powyższych wad nie posiadają protokoły z wiedzą zerową.

- "claimant" nie ujawnia żadnej informacji na temat sekretu,
- wykonanie protokołu nie zwiększa wiedzy weryfikującego,
- weryfikujący nie potrafi powtórzyć protokołu wobec innych stron,
- jedyną informacją przekazywaną weryfikującemu jest "Tak, claimant zna sekret".

Cel protokołu z wiedzą zerową

Udowadniający (ang. *prover*) musi przekonać weryfikującego (ang. *verifier*) o prawdziwości twierdzenia.

- dowody probabilistyczne,
- właściwość kompletności (ang. *completeness*)
 - uczciwy *prover* i *verifier*,
 - prawdopodobieństwo graniczące z pewnością poprawnej identyfikacji/autentykacji,
- właściwość trafności/poprawności (ang. *soundness*),
- właściwość wiedzy zerowej (ang. *zero-knowledge*),
 - uczciwy *prover* i nieuczciwy *verifier*,
 - wykonanie protokołu nie dostarcza wiedzy *verifier*'owi pozwalającej na podszycie się pod *prover*'a

- bezpieczeństwo mimo wielokrotnego stosowania,
- brak szyfrowania,
- większy narzut komunikacyjny i/lub obliczeniowy,
- podobne (te same) podstawy bezpieczeństwa,

Elementy dowodu (Alicja identyfikuje się wobec Boba):

- sekret i opcjonalna informacja publiczna,
- świadek (ang. *witness*) $(A \rightarrow B)$,
- wyzwanie (ang. *challenge*) $(A \leftarrow B)$,
- odpowiedź (ang. *response*) $(A \rightarrow B)$.