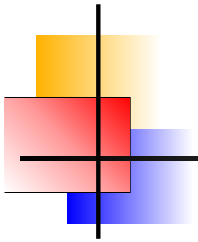
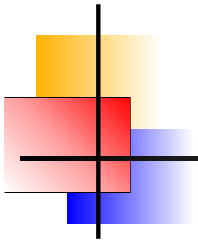


Jednocestné hashovacie funkcie

Stanislav Palúch

Fakulta riadenia a informatiky, Žilinská univerzita

7. decembra 2010



Metódy zabezpečenia správy proti zmene pri prenose

Na kontrolu toho, či správa nebola pri prenose zmenená, sa za správu pridáva časť zložená z kontrolných znakov, pomocou ktorej možno zistiť, či je správa nezmenená.

1. Kontrola paritou
2. Kontrola dekadických kódov modulo 10 resp. modulo 11
3. Lineárne (n, k)-kódy
4. Kontrolný súčet – napr. súčet všetkých čísel správy modulo 2^{64}
5. CRC – cyclic redundancy check
6. ...

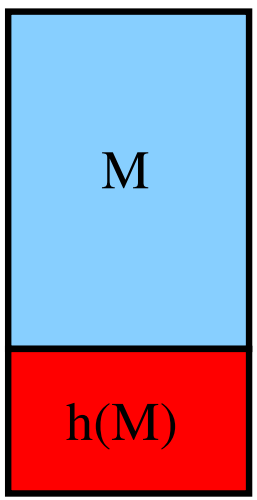
Tieto spôsoby sú účinné proti náhodným chybám, nie však proti zlomyselným útokom.
Útočník totiž ľahko dokáže zmeniť správu tak, tak aby kontrolné znaky zmenenej správy boli rovnaké ako kontrolné znaky pôvodnej správy.

Jednocestné hashovacie funkcie

V kryptografii sa na zaistenie správ proti zmenám pridáva ku každej správe M ďalšia (redundantná) časť, nazývaná **odtlačok správy**.

V anglickej literatúre MAC(M) – Message Autentification Code, MD(M) – Message Digest, Fingerprint.

Tieto sa vypočítavajú pomocou jednocestných hashovacích funkcií.



Požadované vlastnosti hashovacej funkcie $h(M)$

1. Pre každé M je ľahké vypočítať $h(M)$
2. Pre každé h je ťažké nájsť také M, že $h = h(M)$
3. Pre každé M je ťažké nájsť iné M' také, že $h(M) = h(M')$
4. Je ťažké nájsť dve rôzne náhodné správy $M \neq M'$ také, že $h(M) = h(M')$

Dvojica roznych správ M, M' s vlastnosťou $h(M) = h(M')$ sa nazýva kolízia. Vlastnosť 4. sa nazýva odolnosť voči kolízii – collision resistence.

Narodeninový paradox – Birthday paradox

Birthday paradox - narodeninový paradox.

V skupine 23 ľudí sa s pravdepodobnosťou $> \frac{1}{2}$ nájde dvojica, ktorá má v ten istý deň narodeniny.

Majme n možných hodnôt $h(M)$ a náhodne generujme k správ $M_1, M_2, \ldots, M_k$.

Pri prvej správe M_1 nenastane kolízia s pravdepodobnosťou ..... $p = 1$

Pri pridaní druhej správy M_2
nenastane kolízia s pravdepodobnosťou ..... $p = (1 - \frac{1}{n})$

Za predpokladu, že medzi M_1 a M_2 nenastala kolízia,
po pridaní tretej správy M_3
nenastane kolízia s pravdepodobnosťou ..... $p = (1 - \frac{2}{n})$

Za predpokladu, že medzi $M_1, M_2, \ldots, M_{k-1}$ nenastala kolízia,
po pridaní k-tej správy M_k
nenastane kolízia s pravdepodobnosťou ..... $p = (1 - \frac{k-1}{n})$

Pravdepodobnosť kolízie

Pravdepodobnosť, že medzi k správami nenastala ani jedna kolízia je

$$\left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdot\ \ldots\ \cdot\left(1-\frac{k-1}{n}\right)=\prod_{i=1}^{k-1}\underbrace{\left(1-\frac{i}{n}\right)}_{\approx e^{-\frac{i}{n}}}.$$

$$e^{-x}=1-x+\underbrace{\frac{x^2}{2!}-\frac{x^3}{3!}+\frac{x^4}{4!}-\frac{x^5}{5!}+\ldots}_{\text{tento súčet je pre malé } x \text{ zanedbateľný}}$$

$$\prod_{i=1}^{k-1}e^{-\frac{i}{n}}=e^{\sum_{i=1}^{k-1}-\frac{i}{n}}=e^{-\frac{k(k-1)}{2n}}$$

Pravdepodobnosť, že nastala aspoň jedna kolízia je

$$1-e^{-\frac{k(k-1)}{2n}}.$$

Kedy je pravdepodobnosť kolízie $> \varepsilon$?

$$
\begin{aligned}
1 - e^{-\frac{k(k-1)}{2n}} &\geq \varepsilon \\
1 - \varepsilon &\geq e^{-\frac{k(k-1)}{2n}} \\
\ln(1-\varepsilon) &\geq -\frac{k(k-1)}{2n} \\
2n\ln(1-\varepsilon) &\geq -(k^2-k) \\
2n\ln(\frac{1}{1-\varepsilon}) &\leq (k^2-k)
\end{aligned}
$$

$$k^2 - k - 2n\ln(\frac{1}{1-\varepsilon}) = 0$$

$$k_{1,2} = \frac{+1 \pm \sqrt{1 + 4.1.2n\ln(\frac{1}{1-\varepsilon})}}{2} =$$

$$= \frac{1}{2} \pm \sqrt{\frac{1}{4} + 2n\ln(\frac{1}{1-\varepsilon})} \approx \pm\sqrt{2n\ln(\frac{1}{1-\varepsilon})}$$

Veľkosť hashovacej hodnoty

Ak teda $k \geq \sqrt{2n \ln(\frac{1}{1-\varepsilon})}$ pravdepodobnosť kolízie medzi k správami je väčšia než ε.

Pre narodeniny existuje $n = 365$ možností. Položme $\varepsilon = 0.5$ potom

$$k \geq \sqrt{2n \ln(\frac{1}{1-\varepsilon})} = \sqrt{730 \ln(\frac{1}{1-1/2})} == \sqrt{730 \ln(2))} = 22,4944$$

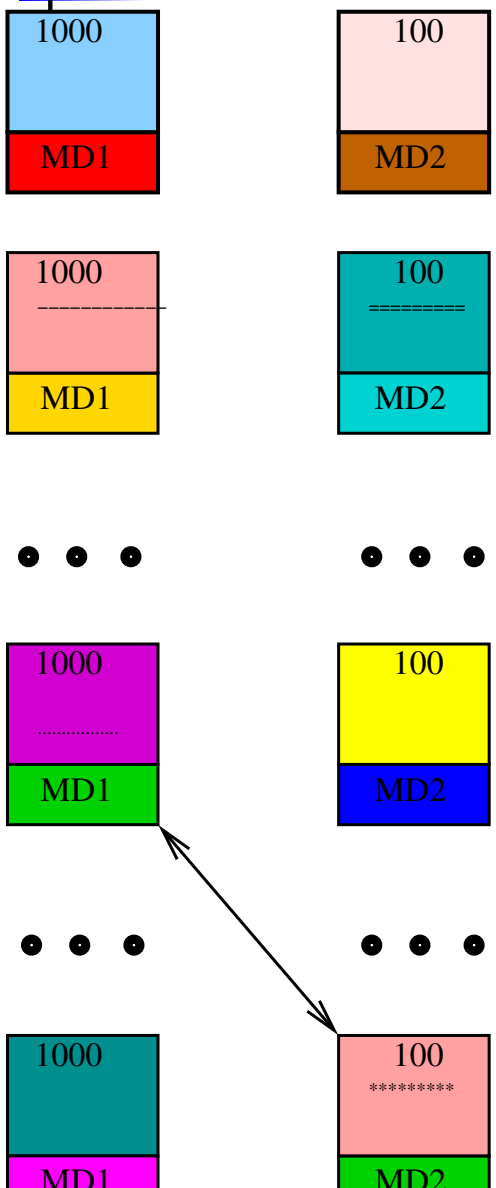
Všeobecne pre n a $\varepsilon = 0.5$

$$k \approx \sqrt{n.2. \ln(\frac{1}{2})} \approx 1,17.\sqrt{n}.$$

Ak by mal odtlačok správy 64 bitov, t.j. $n = 2^{64}$, stačí vytvoriť $1,17 * 2^{32} \approx 5 * 10^9$ náhodných správ, aby sme s pravdepodobnosťou $1/2$ našli kolíziu.

Preto sa používajú odtlačky dlhé 128, 160, 256 bitov.

Birthday attack - Narodeninový útok

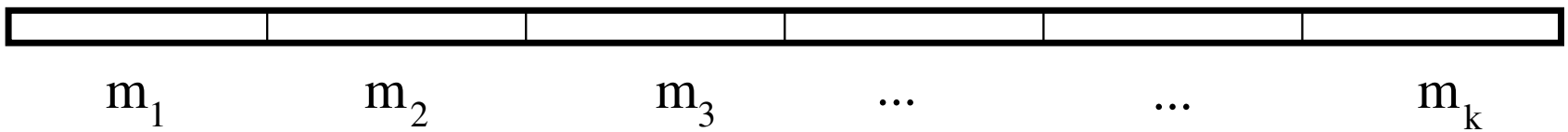


1. Útočník vytvorí dve zmenky – jednu na 100 eur, druhú na 1000 eur
2. Z obidvoch zmeniek bezvýznamnymi zmenami vytvára ich ďalšie varianty dokedy nenájde kolíziu – dvojicu 100 eurového a 1000 eurového variantu s rovnakým odtlačkom h. Ak má odtlačok n možných hodnôt, stačí mu vytvoriť $1{,}17\sqrt{n}$ dvojíc variantov zmeniek, aby s pravdepodobnosťou $> \frac{1}{2}$ našiel kolíziu.
3. Dlžníkovi dá potvrdiť 100 eurový variant s odtlačkom h.
4. Po čase vymáha 1000 eur na základe toho, že mu dlžník potvrdil odtlačok h prislúchajúci 1000 eurovému variantu.

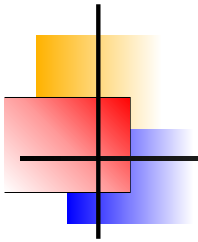
Poučenie: Pred podpisom digitálneho dokumentu vždy v ňom urobiť malú zmenu.

Všeobecný postup tvorby odtlačku správy

1. Správa M, pre ktorú sa robí odtlačok, sa rozdelí na na rovnako dlhé bloky $m_1, m_2, \ldots$.



2. Hashovací algoritmus má pevne stanovený inicializačný vektor IV. Položíme $h_0 = IV$.
3. Rekurzívne počítame $h_i = f(m_i, h_{i-1})$.
4. Výsledný odtlačok celej správy $h(M) = h_k$.



Hash pomocou kryptosystému

$$
\begin{aligned}
h_0 &= IV \\
h_i &= f(m_i, h_{i-1})
\end{aligned}
$$

$$
\begin{aligned}
h_i &= E_{h_{i-1}}(m_i) \oplus m_i \\
h_i &= E_{h_{i-1}}(m_i) \oplus m_i \oplus h_{i-1} \\
h_i &= E_{h_{i-1}}(m_i \oplus h_{i-1}) \oplus m_i \\
h_i &= E_{h_{i-1}}(m_i \oplus h_{i-1}) \oplus m_i \oplus h_{i-1}
\end{aligned}
$$

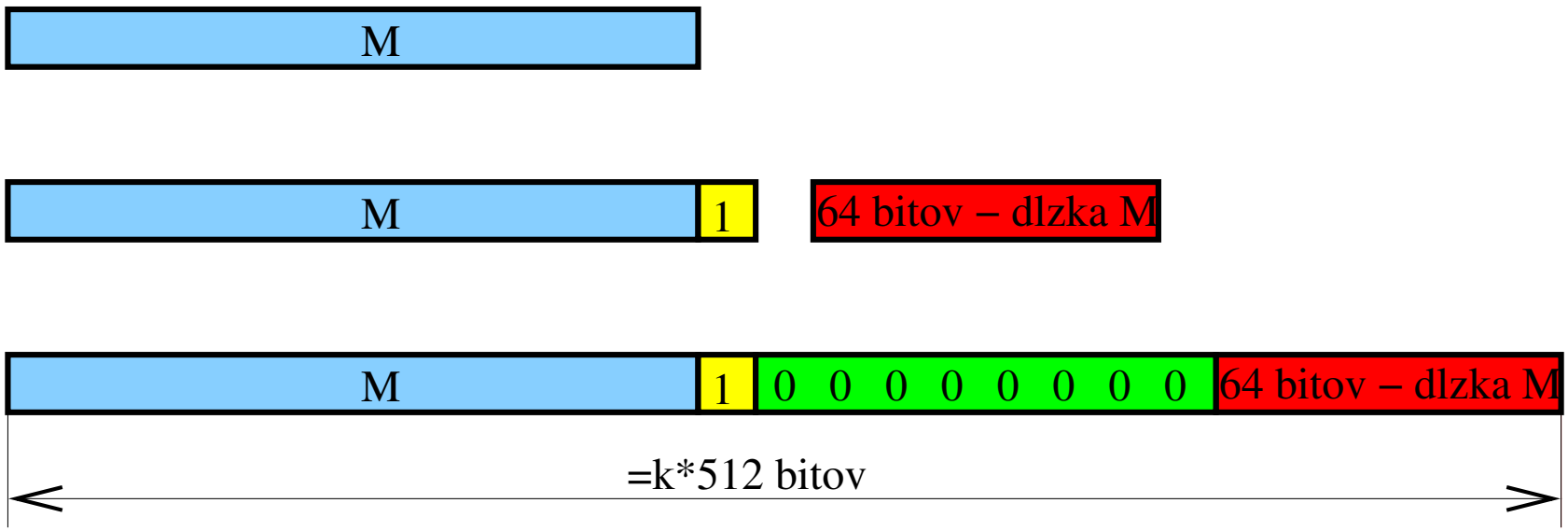
$$
\begin{aligned}
h_i &= E_{m_i}(h_{i-1}) \oplus h_{i-1} \\
h_i &= E_{m_i}(m_i \oplus h_{i-1}) \oplus h_{i-1} \\
h_i &= E_{m_i}(m_i \oplus h_{i-1}) \oplus m_i \oplus h_{i-1} \\
h_i &= E_{m_i}(h_{i-1}) \oplus m_i \oplus h_{i-1}
\end{aligned}
$$

$$
\begin{aligned}
h_i &= E_{m_i \oplus h_{i-1}}(m_i) \oplus m_i \\
h_i &= E_{m_i \oplus h_{i-1}}(h_{i-1}) \oplus h_{i-1} \\
h_i &= E_{m_i \oplus h_{i-1}}(m_i) \oplus h_{i-1} \\
h_i &= E_{m_i \oplus h_{i-1}}(h_{i-1}) \oplus m_i
\end{aligned}
$$

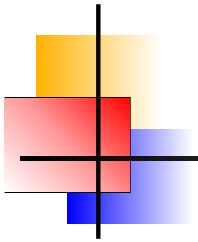
Prelomená schéma $h_i = E_{m_i}(h_{i-1})$.

MD4

Správa sa pred výpočtom odtlačku musí upraviť takto:



1. Pridá sa jeden bit s hodnotou 1.
2. Vytvorí sa 64-bitové číslo obsahujúce dĺžku správy. Týmto číslom bude upravená správa končiť.
3. Medzi doplnenú jednotku a 64 bitov dĺžky sa vloží toľko núl, aby výsledná dĺžka správy bola násobkom 512.



MD4

- Dĺžka odtlačku algoritmu MD4 je 128 bitov, t.j. h_i má 128 bitov.
- S h_i sa pracuje ako so štvoricou (A, B, C, D) 32-bitových čísel
- Spracovávaná dĺžka bloku správy m_i je 512 bitov.
- S blokom textu sa pracuje ako so 16-ticou

$$X[0], X[1], X[2], \ldots, X[15]$$

  32-bitových čísel.
- Inicializačne sa nastaví hodnota $h_0 \equiv (A, B, C, D)$
- i-tý 512-bitový blok textu m_i sa vyjadrí v tvare šestnástich 32-bitových čísel $X[0], X[1], X[2], \ldots, X[15]$ a rekurentne sa vypočíta

$$h_i = f(m_i, h_{i-1})$$

- Ak m_k je posledný blok správy, potom h_k je odtlačok celej správy.

MD4 – funkcia $f(m_i, h_{i-1})$

Význam použitých operácií:

$+$ – sčítanie modulo 2^{32}
$\wedge$ – logické and po bitoch
$\vee$ – logické or po bitoch
$\neg$ – logická negácia po bitoch

MD4 bude používať tieto funkcie:
$f(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$
$g(X, Y, Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$
$h(X, Y, Z) = X \oplus Y \oplus Z$

Nastavenie 128-bitového inicializačného vektora $IV = (A, B, C, D)$:
$A = 67452301$
$B = efcdab89$
$C = 98badcfe$
$D = 10325476$

Funkcia $h_i = f(m_i, h_{i-1})$

0 Vstup $(A, B, C, D) = h_{i-1}$,
$(X[0], X[1], \ldots, X[15]) = m_i$

1. Uloženie $h_{i-1} = (A, B, C, D)$
$AA = A$
$BB = B$
$CC = C$
$DD = D$

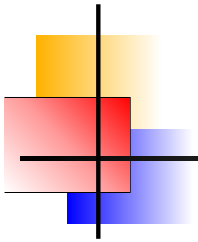
2. 1.kolo$(A, B, C, D, X[0-15])$

3. 2.kolo$(A, B, C, D, X[0-15])$

4. 3.kolo$(A, B, C, D, X[0-15])$

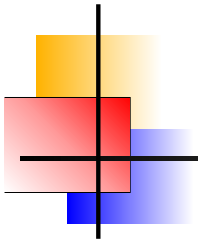
5. $A = A + AA$
$B = B + BB$
$C = C + CC$
$D = D + DD$

6 Return $h_i = (A, B, C, D)$



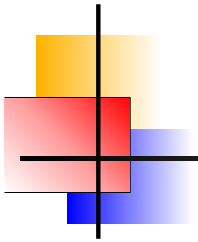
1. kolo MD4

$$
\begin{aligned}
1.\quad & A = (A + f(B,C,D) + X[0]) <<< 3 \\
2.\quad & D = (D + f(A,B,C) + X[1]) <<< 7 \\
3.\quad & C = (C + f(D,A,B) + X[2]) <<< 11 \\
4.\quad & B = (B + f(C,D,A) + X[3]) <<< 19 \\
5.\quad & A = (A + f(B,C,D) + X[4]) <<< 3 \\
6.\quad & D = (D + f(A,B,C) + X[5]) <<< 7 \\
7.\quad & C = (C + f(D,A,B) + X[6]) <<< 11 \\
8.\quad & B = (B + f(C,D,A) + X[7]) <<< 19 \\
9.\quad & A = (A + f(B,C,D) + X[8]) <<< 3 \\
10.\quad & D = (D + f(A,B,C) + X[9]) <<< 7 \\
11.\quad & C = (C + f(D,A,B) + X[10]) <<< 11 \\
12.\quad & B = (B + f(C,D,A) + X[11]) <<< 19 \\
13.\quad & A = (A + f(B,C,D) + X[12]) <<< 3 \\
14.\quad & D = (D + f(A,B,C) + X[13]) <<< 7 \\
15.\quad & C = (C + f(D,A,B) + X[14]) <<< 11 \\
16.\quad & B = (B + f(C,D,A) + X[15]) <<< 19
\end{aligned}
$$



2. kolo MD4

$$
\begin{aligned}
1.\ & A &=& (A + g(B,C,D) + X[0] + 5a827999) <<< 3 \\
2.\ & D &=& (D + g(A,B,C) + X[4] + 5a827999) <<< 5 \\
3.\ & C &=& (C + g(D,A,B) + X[8] + 5a827999) <<< 9 \\
4.\ & B &=& (B + g(C,D,A) + X[12] + 5a827999) <<< 13 \\
5.\ & A &=& (A + g(B,C,D) + X[1] + 5a827999) <<< 3 \\
6.\ & D &=& (D + g(A,B,C) + X[5] + 5a827999) <<< 5 \\
7.\ & C &=& (C + g(D,A,B) + X[9] + 5a827999) <<< 9 \\
8.\ & B &=& (B + g(C,D,A) + X[13] + 5a827999) <<< 13 \\
9.\ & A &=& (A + g(B,C,D) + X[2] + 5a827999) <<< 3 \\
10.\ & D &=& (D + g(A,B,C) + X[6] + 5a827999) <<< 5 \\
11.\ & C &=& (C + g(D,A,B) + X[10] + 5a827999) <<< 9 \\
12.\ & B &=& (B + g(C,D,A) + X[14] + 5a827999) <<< 13 \\
13.\ & A &=& (A + g(B,C,D) + X[3] + 5a827999) <<< 3 \\
14.\ & D &=& (D + g(A,B,C) + X[7] + 5a827999) <<< 5 \\
15.\ & C &=& (C + g(D,A,B) + X[11 + 5a827999]) <<< 9 \\
16.\ & B &=& (B + g(C,D,A) + X[15] + 5a827999) <<< 13
\end{aligned}
$$

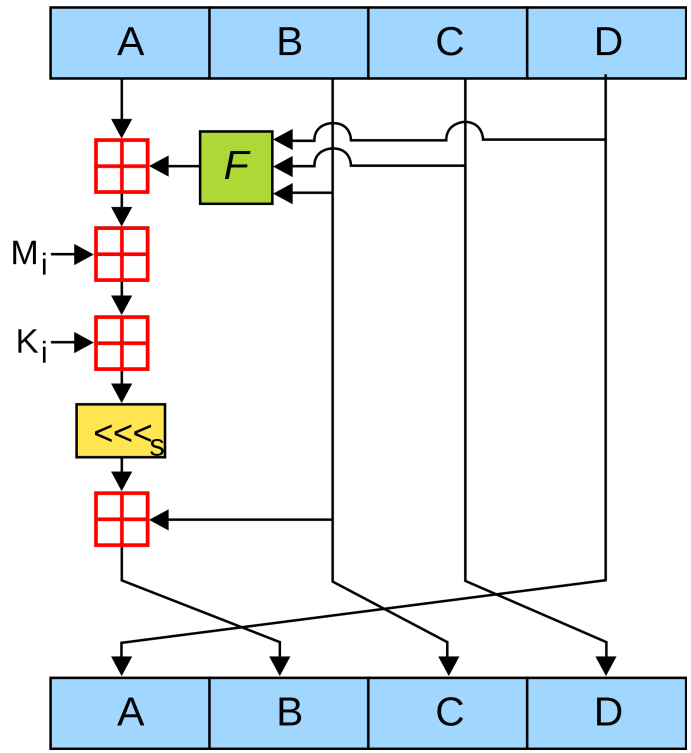


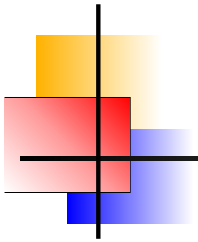
3. kolo MD4

$$
\begin{aligned}
1.\quad & A &=& \ (A + h(B,C,D) + X[0] \ + 6ed9eba1) <<< 3 \\
2.\quad & D &=& \ (D + h(A,B,C) + X[8] \ + 6ed9eba1) <<< 9 \\
3.\quad & C &=& \ (C + h(D,A,B) + X[4] \ + 6ed9eba1) <<< 11 \\
4.\quad & B &=& \ (B + h(C,D,A) + X[12] + 6ed9eba1) <<< 15 \\
5.\quad & A &=& \ (A + h(B,C,D) + X[2] \ + 6ed9eba1) <<< 3 \\
6.\quad & D &=& \ (D + h(A,B,C) + X[10] + 6ed9eba1) <<< 9 \\
7.\quad & C &=& \ (C + h(D,A,B) + X[6] \ + 6ed9eba1) <<< 11 \\
8.\quad & B &=& \ (B + h(C,D,A) + X[14] + 6ed9eba1) <<< 15 \\
9.\quad & A &=& \ (A + h(B,C,D) + X[1] \ + 6ed9eba1) <<< 3 \\
10.\quad & D &=& \ (D + h(A,B,C) + X[9] \ + 6ed9eba1) <<< 9 \\
11.\quad & C &=& \ (C + h(D,A,B) + X[5] \ + 6ed9eba1) <<< 11 \\
12.\quad & B &=& \ (B + h(C,D,A) + X[13] + 6ed9eba1) <<< 15 \\
13.\quad & A &=& \ (A + h(B,C,D) + X[3] \ + 6ed9eba1) <<< 3 \\
14.\quad & D &=& \ (D + h(A,B,C) + X[11] + 6ed9eba1) <<< 9 \\
15.\quad & C &=& \ (C + h(D,A,B) + X[7] \ + 6ed9eba1) <<< 11 \\
16.\quad & B &=& \ (B + h(C,D,A) + X[15] + 6ed9eba1) <<< 15
\end{aligned}
$$

MD5

1. Je zosilnením algoritmu MD4.
2. Dáva 128-bitový hash.
3. Pracuje s 512-bitovým blokom textu
4. Namiesto troch kôl má 4 kolá
5. Má pozmenené funkcie takto
   $F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$
   $G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$
   $H(X, Y, Z) = X \oplus Y \oplus Z$
   $I(X, Y, Z) = Y \oplus (X \vee \neg Z)$
6. Beží asi o 30% pomalšie než MD4



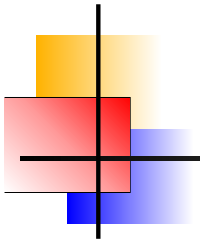


SHA algoritmus

1. SHA produkuje 160-bitový hash
2. Spracováva 512 bitový blok textu $W[0], W[1], \ldots, W[15]$, ktorý expanduje do 80 takto: pre $16 \geq j \leq 79$

$$W[j] = W[j-3] \oplus W[j-8] \oplus W[j-14] \oplus W[j-16]$$

3. Má 4 kolá po 20 krokov



SHA-1

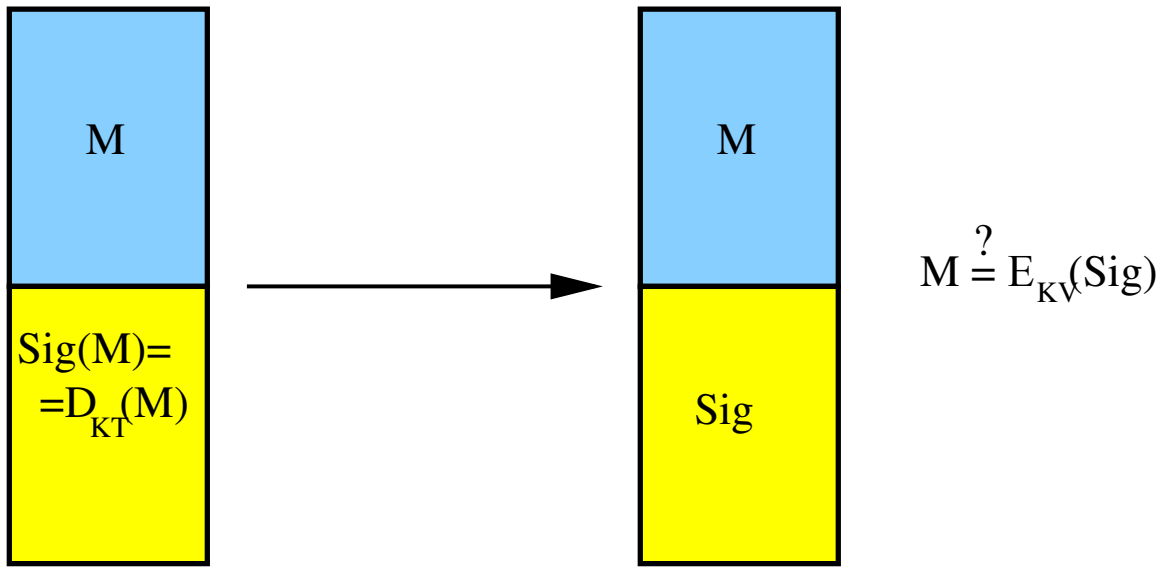
```
Initialize hash value for this chunk:
a = h0; b = h1; c = h2; d = h3; e = h4;
Main loop:
for( i=0; i < 80; i++)
  {if (0 <= i) and (i <= 19) then   {f = (b and c) or ((not b) and d;)
                                     k = 0x5A827999;}
  else if( 20<=i) and (i <= 39)     {f = b xor c xor d;
                                     k = 0x6ED9EBA1;}
  else if(40 <= i) and (i <= 59) {f = (b and c) or (b and d)or(c and d);
                                     k = 0x8F1BBCDC;}
  else if(60<= i) and (i <= 79)   {f = b xor c xor d;
                                     k = 0xCA62C1D6;}
  temp = (a leftrotate 5) + f + e + k + w[i];
  e = d; d = c; c = b leftrotate 30; b = a; a = temp;
}

  Add this chunk's hash to result so far:
  h0 = h0 + a; h1 = h1 + b; h2 = h2 + c; h3 = h3 + d; h4 = h4 + e;
```

Digitálny podpis

- Účastník A s dvojicou kľúčov KV_A, KT_A podpíše správu M tak, že k nej pripojí výsledok dešifrovania správy M kľúčom KT_A. Teda

$$Sig(M) = D_{KT_A}(M).$$



- Účastník B overí pravosť podpisu tak, že vypočíta $M' = E_{KV_A(Sig(M))}$ a skontroluje, či $M = M'$.

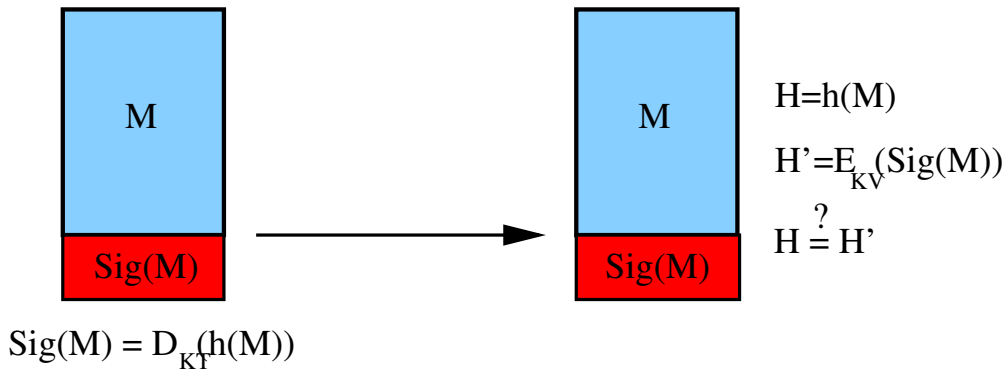
  Ak $M' \neq M$, potom buď správa bola zmenená, alebo podpis nie je pravý.

  Ak $M' = M$, potom je podpis pravý a správa nezmenená.

  Jediný človek – účastník A – mohol ku správe M vytvoriť $Sig(M) = D_{KT_A}(M)$, pretože on jediný má kľúč KT_A.

Digitálny podpis

- Účastník A s dvojicou kľúčov KV_A, KT_A podpíše správu M tak, že
  1. Vypočíta $h(M)$ odtlačok správy M.
  2. Odtlačok správy $h(M)$ zašifruje svojim tajným kľúčom:
     $$Sig(M) = D_{KT_A}(h(M)).$$
  3. $Sig(M)$ pripojí k správe M ako svoj digitálny podpis



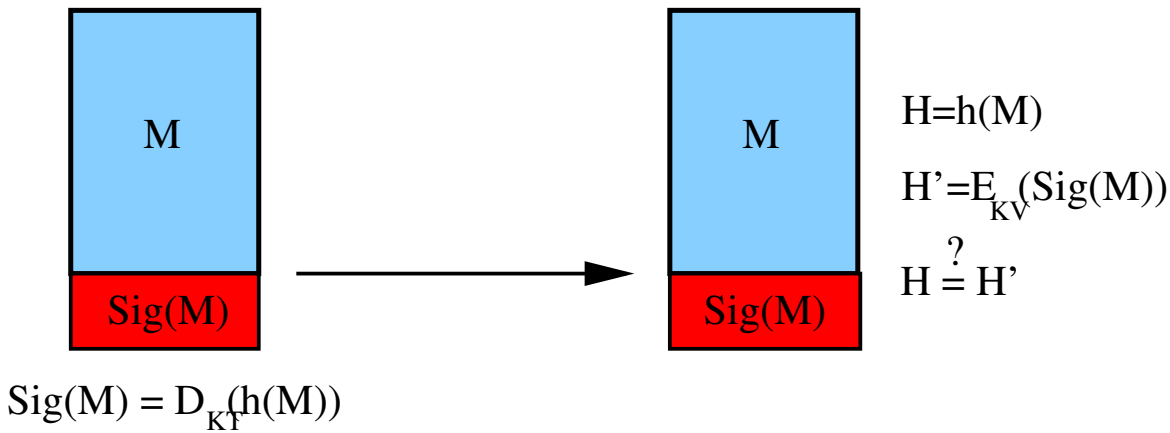
- Účastník B overí pravosť podpisu tak, že
  1. Vypočíta $H = h(M)$
  2. Vypočíta $H' = E_{KV_A}(h(M))$.
  3. Skontroluje, či $H' = H$.

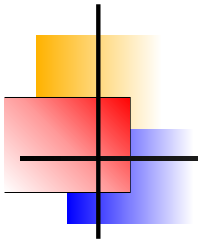
     Ak $H' \neq H$, potom buď správa bola zmenená, alebo podpis nie je pravý.

     Ak $H' = H$, potom je podpis pravý a správa nezmenená.

     Jediný človek – účastník A – mohol ku správe M vytvoriť $Sig(M) = D_{KT_A}(h(M))$, pretože on jediný má kľúč KT_A.

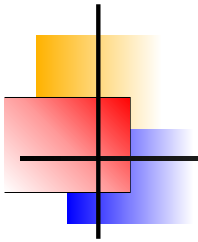
Digitálny podpis





Časové značky

- K správe M pridáme časť $X = MD(M), PUB$, kde $MD(M)$ je odtlačok správy M a PUB je verejne známa informácia, ktorá vznikla v deň podpisu správy M.
- Ako podpis pridáme časť $Y = sig(MD(M), PUB)$.
- V novinách publikujeme trojicu $MD(M)$, PUB, $sig(MD(M), PUB)$, takže môžeme dokázať, že sme podpísali správu s odtlačkom $MD(M)$, s verejne známou informáciou PUB.



Jednocestné funkcie

Máme rovnicu v obore reálnych čísel s neznámou x

$$2^x = a.$$

Jej riešením je $x = \log_2(a)$.

Podobne riešenie rovnice $z^x = a$, kde $z > 0$ a x neznáma, je $x = \log_z(a)$.

Diskrétny logaritmus

Poznám a, prvočíslo p a číslo s: $1 < s < p$. Na aké x musím umocniť s, aby

$$s^x = a \mod p?$$

Iná formulácia. Aké je riešenie rovnice

$$s^x = a$$

v poli $\mathbb{Z}^p$?

Tento problém sa nazýva problém diskrétneho logaritmu.
Pre veľké p je hľadanie diskrétneho logaritmu ťažká úloha.

Diffie - Hellmanova výmena kľúčov

A a **B** sa dohodnú na veľkom prvočísle p a čísle s, $1 < s < p$.

Čísla s, p môžu byť verejné, použiteľné opakovane aj pre viac používateľov.

A

- Zvolí $a < p$ tajné.
- Vypočíta $\alpha = s^a \mod p$.
- Odošle α.
- Prijme β.
- Vypočíta kľúč $K_A = \beta^a \mod p$

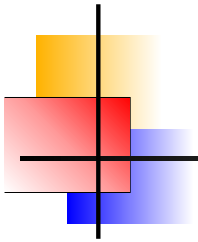
B

- Zvolí $b < p$ tajné.
- Vypočíta $\beta = s^b \mod p$.
- Odošle β.
- Prijme α.
- Vypočíta kľúč $K_B = \alpha^b \mod p$

Je $K_A = K_B$?

Platí:

$$K_A = \beta^{\sim} = \left(s^b\right)^a = s^{ab} = (s^a)^b = \alpha^b = K_B \mod p$$



Diffie - Hellmanova výmena kľúčov

Nebezpečenstvo: Intruder in the middle attacack

$$\mathbf{A} \xrightarrow{\alpha=s^a} \mathbf{X} \xrightarrow{\alpha'=s^{a'}} \mathbf{B}$$

$$\mathbf{A} \xleftarrow{\beta'=s^{b'}} \mathbf{X} \xleftarrow{\beta=s^b} \mathbf{B}$$

$$\mathbf{A} \xleftrightarrow{K_1=s^{ab'}} \mathbf{X} \xleftrightarrow{K_2=s^{a'b}} \mathbf{B}$$

Prístupové heslá

Používateľ	Počítač
heslo	Skontroluje, či sa zhoduje s uloženým heslom
heslo	Skontroluje, či sa zhoduje s uloženým MD hesla

Slovníkový útok - Dictionary attack

- krstné mená
- zemepisne názvy
- astronomické názvy
- bájne postavy
- biblické postavy
- chemické prvky
- dni a mesiace
- mená hercov, umelcov, spevákov
- názvy kníh, diel, udalostí

Používateľ	Salt	MD(Heslo, Salt)
peterp	$EA1DFC48_H$	128 bitov