Rozšířená verze příspěvku, který byl zaslán do mezinárodní kryptologické konference EUROCRYPT 2007.

Nový koncept hašovacích funkcí SNMAC s využitím speciální blokové šifry a konstrukcí NMAC/HMAC

Vlastimil KLÍMA[*]
říjen 2006[**]

Abstrakt. V příspěvku prezentujeme nové důkazy bezpečnosti velmi dobře známých hašovacích konstrukcí NMAC/HMAC, navržené Bellare a kol. v roce 1996. Ukazujeme, že blokové šifry by měly být v hašovacích funkcích používány jiným způsobem než dosud. Zavádíme nové kryptografické primitivum, speciální blokovou šifru (SBŠ). SBŠ je odolná proti útokům, specifickým pro blokové šifry v hašovacích funkcích. Navrhujeme nový koncept hašovacích funkcí (SNMAC, Special NMAC), který vzniká použitím SBŠ v konstrukcích NMAC/HMAC. Z nových důkazů bezpečnosti NMAC/HMAC vyplývá, že hašovací funkce SNMAC jsou výpočetně odolné proti nalezení vzoru a kolize. Navíc Coron a kol. na CRYPTO 2005 ukázali, že SNMAC se limitně blíží náhodnému orákulu. Konstrukce SNMAC je obecná a umožňuje různorodé návrhy pomocí různých instancí SBŠ. Navrhujeme speciální blokovou šifru DN (Double Net) a na základě ní konstruujeme hašovací funkci HDN (Hash Double Net) jako konstrukci typu SNMAC.

Obsah

[*] nezávislý konzultant, v.klima (at) volny.cz, http://cryptography.hyperlink.cz, V tomto příspěvku prezentujeme část projektu NBÚ Bezpečná hašovací funkce (ST20052005017).
[**] druhá verze tohoto příspěvku bude obsahovat Dodatky 2 a 3 (čekají na schválení publikace)

1. Úvod

Je známo, že většina používaných iterativních hašovacích funkcí podléhá tzv. útoku prodloužením zprávy [Tsu92], tj. z hodnoty h(M) lze vypočítat h(M || N) pro vhodné N. I když se tím odlišují od náhodného orákula, tato vlastnost byla mnoha hašovacím funkcím dlouho tolerována. V roce 2004 a 2005 byly zjištěny další generické problémy hašovacích funkcí, multikolizní útok Jouxové [Jou04] a Kelsey-Schneierův multikolizní útok a útok na druhý vzor [KS05]. Poznamenejme, že všechny moderní hašovací funkce podléhají těmto třem generickým útokům ([Tsu92], [Jou04], [KS05]), a proto se silně odlišují od chování náhodného orákula.

Jako možná náhrada funkcí MD5 a SHA-1 byla dříve uvažována třída SHA-2 [SHA-2]. Hašovací funkce této třídy však také mají všechny generické slabiny a navíc jejich návrhová kritéria nebyla nikdy publikována. Avšak i u této třídy hašovacích funkcí se začínají objevovat útoky. Využívají slabých nelinearit v expanzi klíče a v použité blokové šifře ([HPR04], [SKH04], [YB05], [YBP05], [MPRR06a], [MPRR06b]).

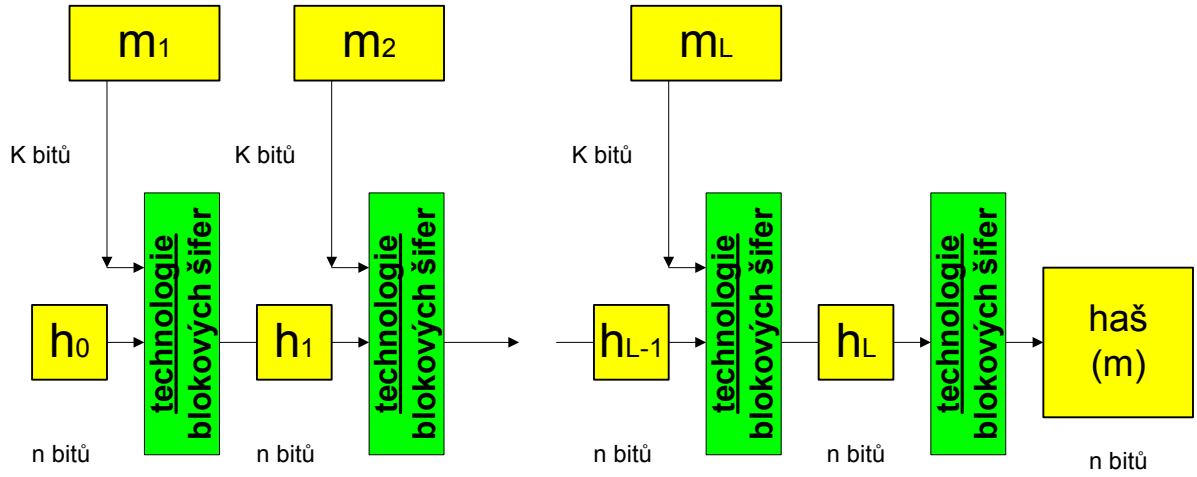
Také praktická kryptoanalýza hašovacích funkcí v posledních letech velmi pokročila. Byly odhaleny závažné slabiny v řadě hašovacích funkcí ([WY05], [WYY05a], [WYY05b], [YWYP06], [BCJ05], [Kli06a]), zejména u MD5, SHA-0 a SHA-1 ([MD5], [SHA-0], [SHA-1]). Generické útoky na současné nejsilnější hašovací funkce ([Tsu92], [Jou04], [KS05]) a praktické útoky na funkce třídy MD a SHA ukázaly, že je potřeba navrhnout novou filozofii hašovacích funkcí ([Sch04]).

Konstrukce NMAC/HMAC navrhli v roce 1996 Bellare a kol. na CRYPTO 1996 [BCK96]. Coron a kol. [CDMP05] na CRYPTO 2005 zkoumali konstrukci typu NMAC se dvěma náhodnými orákuly a konstrukci typu HMAC s ideální blokovou šifrou v Davies-Meyerově úpravě. Dokázali, že se zvyšováním délky bloku se tyto konstrukce stávají neodlišitelné od náhodných orákul. V tomto příspěvku poprvé dokazujeme kvantitativní odhady odolnosti uvedených konstrukcí proti nalezení vzoru a kolize. Z Věty 1 až 4 vyplývá, že pro nalezení kolize NMAC/HMAC je útočník nucen vykonat řádově $2^{n/2}$ operací a pro nalezení vzoru NMAC/HMAC řádově 2^n operací, tedy stejně jako u náhodných orákul. Konstrukce NMAC/HMAC, které navrhli v roce 1996 Bellare a kol. [BCK96], se tak stávají prakticky i teoreticky podloženými kandidáty na hašovací funkce nové generace.

V současné době Bellare [Bel06] ukázal, že v konstrukci HMAC je dokonce možné zeslabit tradiční požadavky na kompresní funkci. Například k tomu, aby HMAC byla PRF postačuje, aby kompresní funkce byla PRF.

Protože funkce NMAC/HMAC jsou výpočetně odolné proti nalezení vzoru a kolize, nemusíme se obávat generických útoků, které objevili Jouxová a Kelsey-Schneier ([Jou04], [KS05]). Zbývajícím generickým útokem je útok rozšířením zprávy. V současné práci Gauravarama a kol. [GHA06] se takový útok ukazuje pro NMAC na základě pouze velmi vyumělkovaného tvaru jeho vnitřních funkcí.

Druhá část příspěvku se zabývá praktickou konstrukcí funkcí NMAC/HMAC a návrhem hašovací funkce SNMAC. Příčinou současných útoků na hašovací funkce tříd MD a SHA, včetně SHA-2, jsou slabé nelinearity v použité blokové šifře a její expanzi klíče. Aby se nové hašovací funkce vyhnuly moderním útokům ([KML02], [BDK03], [HKK03], [KKH04], [SKH04], [KKL04], [BDK05], [HKL05], [KBP05], [MPRR06a], [MPRR06b], [YWYP06], [BDK07]), měly by odstranit tyto slabě nelineární funkce ze svých návrhů a nahradit je současnou *technologií* blokových šifer [Bih05]. Technologií máme na mysli osvědčené principy a stavební bloky blokových šifer. Pokud tuto technologii použijeme, dostáváme hašovací funkci na obr. 1.



Obr. 1: Hašovací funkce na bázi technologie blokových šifer

Ukazujeme, že blokové šifry by měly být použity v hašovacích funkcích jiným způsobem než dosud. Nazýváme je speciální blokové šifry (SBŠ) a formulujeme jejich vlastnosti. Toto nové kryptografické primitivum se vymyká klasické představě blokových šifer. Základní vlastností SBŠ je, že útočník má plnou kontrolu nad jejím klíčem. S takovým požadavkem nebyly dosud blokové šifry konstruovány, a proto žádné současné blokové šifry nejsou příliš vhodné pro použití v hašovacích funkcích. Konstrukci blokové šifry, která bude použita v hašovacích funkcích, je nutné podřídit požadavku, že útočník má plnou kontrolu nejen nad otevřeným a šifrovým textem, ale i nad klíčem. Není to pouze teoretický koncept, praktický příklad SBŠ uvádíme v [Kli06b].

Důkazy bezpečnosti konstrukcí NMAC/HMAC jsou založeny na faktu, že f a g jsou nezávislá náhodná orákula v NMAC a E je ideální bloková šifra v HMAC. Pokud máme k dispozici speciální blokovou šifru, můžeme ji přímo použít jako náhodné orákulum v modelu NMAC (viz. obr. 3), který je obecnější než HMAC (viz obr. 4). Tuto konstrukci nazýváme SNMAC (Special NMAC) podle toho, že používá speciální blokovou šifru (SBC) v modelu NMAC. Poznamenejme, že model HMAC využívá klasickou blokovou šifru. Pokud bychom ji smysluplně přeměnili na speciální blokovou šifru, obdrželi bychom SNMAC také. Konstrukce SNMAC je tedy jakýmsi kompromisem mezi HMAC a NMAC. Dostaneme se k ní zdola "zesílením" HMAC (použitím speciální blokové šifry namísto klasické) nebo shora "zeslabením" NMAC (použitím speciální blokové šifry namísto náhodných orákul).

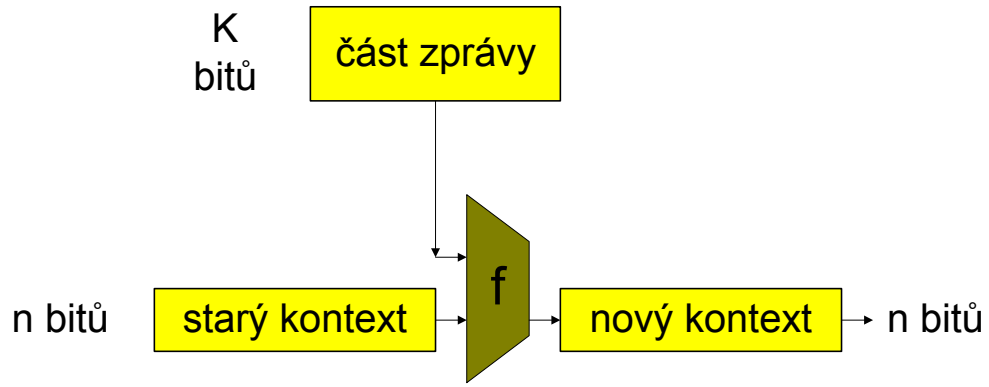
Navrhujeme koncept SNMAC jako kandidáta na hašovací funkci nové generace. Je výpočetně odolný proti nalezení vzoru a kolize, limitně se blíží náhodnému orákulu a umožňuje návrh různých instancí pomocí různých SBŠ.

SNMAC využívá speciální blokovou šifru zvláštním způsobem v kompresní funkci, a to podle vztahu $h_i = SBŠ_{hi-1 \,||\, mi}(Const_0)$.Využívá faktu, že dlouhá desetiletí byly blokové šifry konstruovány tak, aby ze znalosti libovolného množství otevřeného a šifrového textu nebylo možné určit klíč. Tím se konstrukce SNMAC brání proti určení vzoru, neboť ten vstupuje do klíče SBŠ. Dále využívá vlastnosti, že při pevném otevřeném textu a proměnném klíči jsou šifrové texty náhodné a příslušné zobrazení je náhodné orákulum. Kompresní funkce je pak tímto náhodným orákulem.

Příspěvek má následující obsah. V kapitole 2 je uvedena definice NMAC a HMAC, v kapitole 3 hlavní věty o odolnosti NMAC a HMAC vůči nalezení vzoru a kolize. V kapitole 4 uvádíme koncept SBŠ a SNMAC. Konkrétní instance SBŠ a SNMAC je popsána v kapitole 5 a příspěvek uzavíráme v kapitole 6. V dodatku 1 jsou obsaženy důkazy hlavních vět z kapitoly 3. Dodatky 2 a 3 obsahují definice funkcí DN a HDN.

2. Definice NMAC a HMAC

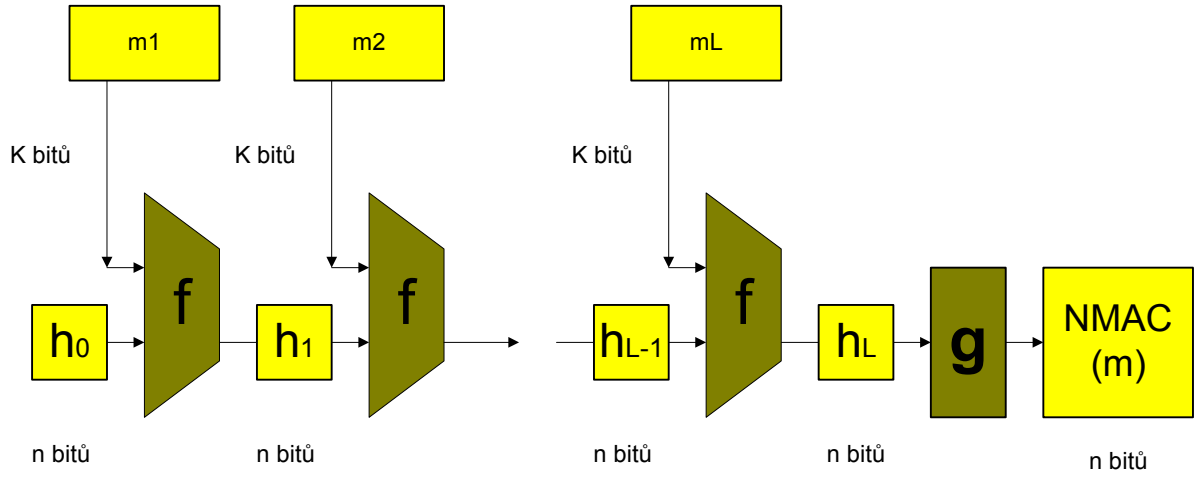
Základní konstrukce. V praxi se setkáváme s nutností hašovat zprávy po částech. Například když z komunikačního kanálu dostáváme zprávu jako posloupnost a nemáme dostatek paměti na uložení celého proudu. Představme si hašovací funkci jako konečný automat. Po zpracování určité části zprávy dostáváme jako výsledek vnitřní stav tohoto automatu, který u hašovací funkce nazýváme kontext. Vstupem do dalšího kroku konečného automatu je tento kontext a další část zprávy. První kontext konečného automatu označujeme jako inicializační hodnota. Dostáváme tak základní model, využívající kompresní funkci f, viz obr. 2. Z přirozeného požadavku, aby kompresní funkce f byla definována pro konstantní šířku vstupu, dostáváme nutnost zarovnání zprávy a její dělení na stejné bloky. Tím obdržíme klasický Merkle-Damgardův model iterativní hašovací funkce, který je základem všech moderních hašovacích funkcí [Mer89][Dam89].



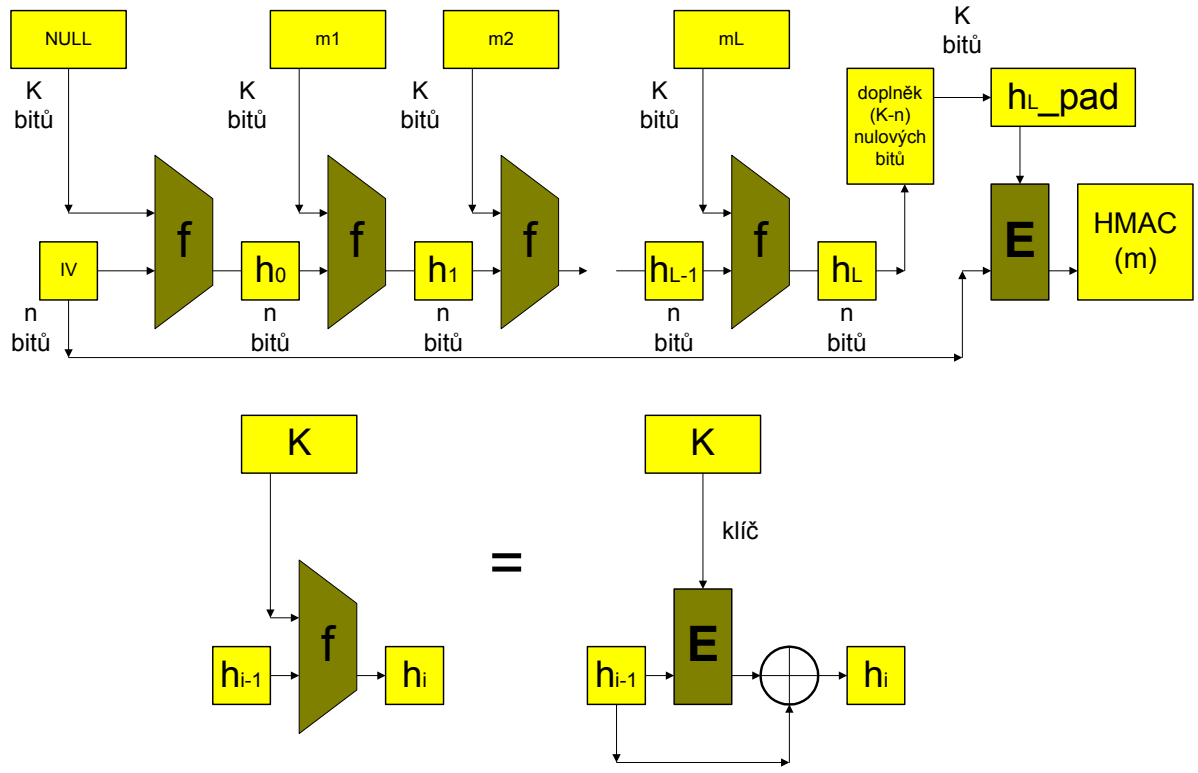
Obr.2: Iterativní hašovací funkce

Bohužel právě tento model má tři uvedené generické slabiny, a to nezávisle na tom, jaký je obsah kompresní funkce f. Zejména umožňuje nalézat multikolize a multivzory s menším úsilím než u náhodného orákula ([Jou04], [KS05]). Opustit velmi přirozenou konstrukci na bázi iterativního principu ([Mer89], [Dam89], [BCK96]) však nemůžeme. Proto se musíme smířit s tím, že hašovací funkce nové generace bude z teoretického hlediska náchylná k multikolizním útokům. Vhodnou obranou může být výpočetní složitost. Funkce musíme konstruovat tak, aby tyto útoky vyžadovaly příliš mnoho operací. V konstrukci na obr. 3 a 4 je použita závěrečná operace g. To je obrana proti třetímu generickému útoku - útoku prodloužením zprávy. Nezabrání mu teoreticky ale učiní jej velmi nepravděpodobným v praxi.

Protože funkce f a g jsou různé, nepůjde k tvorbě h(M || N) použít h(M) jednoduše. Výpočet h(M) končí operací g, zatímco při výpočtu h(M || N) je v příslušném místě použita operace f. Pokud použijeme dvě náhodná orákula f a g, dostáváme konstrukci NMAC (viz obr. 3) podle [BCK96], [CDMP05]. Pokud tato orákula konstruujeme pomocí blokové šifry například v Davies-Meyerově formě [MMO85], dostáváme konstrukci, kterou označujeme HMAC (viz obr. 4) podle [BCK96], [CDMP05]. Poznamenáváme, že se jedná formálně o mírně odlišnou definici HMAC, než která je standardizovaná například v [RFC2104].



Obr.3: Definice hašovací funkce NMAC (viz [BCK96], [CDMP05])



Obr.4: Definice hašovací funkce HMAC (viz [BCK96], [CDMP05])

U obou modelů HMAC a NMAC uvažujeme, že zpráva se standardně doplňuje (bitem 1, bity 0 a délkou původní zprávy) a zarovnává se na bloky stejné délky (K bitů) podobně jako u SHA-2.

3. Bezpečnost hašovacích funkcí HMAC a NMAC

V této kapitole dokážeme věty o odolnosti HMAC a NMAC proti nalezení kolize a vzoru. Věty 1 až 4 obsahují kvantitativní odhady pravděpodobnosti nalezení kolize nebo vzoru v závislosti na počtu operací, které má útočník k dispozici. Obdržené odhady jsou velmi těsné, neboť dolní a horní meze jsou řádově stejné.

Z Věty 1 až 4 vyplývá, že pro nalezení kolize je útočník nucen vykonat řádově $2^{n/2}$ operací a pro nalezení vzoru řádově 2^n operací, tedy hašovací funkce HMAC a NMAC se vůči němu v těchto případech chovají podobně jako náhodná orákula.

V dalším použijeme obvyklou definici black-box modelu blokové šifry podle [BRS02].

3.1. Věty o bezpečnosti HMAC

Označme BC(K, n) množinu všech blokových šifer E, které mají K bitový klíč a n bitový blok. Nechť E je náhodně vybraná bloková šifra z množiny BC(K, n), tj. $E \xleftarrow{\$} BC(K, n)$, kde symbol $\xleftarrow{\$} M$ označuje náhodný výběr objektu z množiny M.

Model black-boxu ([BRS02], str. 322). Tento model se připisuje Shannonovi a byl použit v pracích jako [W84], [KR96], [EM91]. Nechť K je pevná délka klíče a n délka bloku blokové šifry E. Útočník A má přístup k orákulům E a E^{-1}, kde E je náhodná bloková šifra E: $\{0, 1\}^K$ x $\{0, 1\}^n \rightarrow \{0, 1\}^n$ a E^{-1} její inverze. To znamená, že pro každý klíč $k \in \{0, 1\}^K$ je $E_k = E(k, *)$ náhodně vybraná permutace na množině $\{0, 1\}^n$ a útočník má přístup k orákulům E a E^{-1}. Tohoto útočníka, tj. jakýkoliv algoritmus útoku, který má přístup k orákulům E nebo E^{-1}, označujeme A_E_E^{-1}. Poznamenejme, že pro vstup (k, y) orákulum E^{-1} vrací hodnotu x takovou, že $y = E_k(x)$. A_E_$E^{-1}(\sigma)$ označuje algoritmus, zvolený na základě parametru σ.

I když HMAC bude použita s $K \geq n$, některé důkazy níže platí i pro $K < n$. Při závěrečné úpravě u HMAC je proto potřeba vzniklou n-bitovou proměnnou h_L doplnit na K bitovou hodnotu. Tuto operaci značíme h_L_pad a označuje doplnění h_L o K - n nulových bitů.
Označme $HMAC^E$ hašovací funkci HMAC, založenou na blokové šifře E, viz obr. 4.

Konvence ([BRS02], str. 328). V dalším budeme uvažovat následující významné předpoklady. Za prvé, útočník se neptá orákula na žádnou otázku, na kterou již zná odpověď. Zejména když se A ptá na Ek(x) a obdrží odpověď y, pak se už neptá na $E_k(x)$ ani na $E^{-1}{}_k(y)$. Jestliže se A ptal na $E^{-1}{}_k$ (y) a obdržel odpověď x, pak se už neptá na $E^{-1}{}_k(y)$ ani na $E_k(x)$.
Za druhé, když útočník hledající kolizi pro HMAC jako výstup předkládá M a M´, pak se předpokládá, že musel projít výpočtem HMAC(M) a HMAC(M´) v tom smyslu, že se musel zeptat na všechny hodnoty E (nebo E^{-1}), které jsou použité ve všech iteracích během výpočtu HMAC(M) a HMAC(M´). Podobně, když útočník hledající vzor HMAC jako výstup předkládá zprávu M, předpokládáme, že musel projít výpočtem HMAC(M), tj. musel zjistit všechny hodnoty E (nebo E^{-1}), které jsou použité ve všech iteracích během výpočtu HMAC(M).
Důkazy Vět 1 až 4 jsou uvedeny v Dodatku 1.

3.2. Odolnost HMAC proti nalezení vzoru

Věta 1: Odolnost HMAC proti nalezení vzoru.

Nechť $Pr\_E\_\sigma = Pr[E \xleftarrow{\$} BC(K, n); \sigma \xleftarrow{\$} \{0, 1\}^n; M \longleftarrow A\_E\_E^{-1}(\sigma): HMAC^E(M) = \sigma]$ je pravděpodobnost jevu, že pro nějakou náhodně vybranou hodnotu haše σ a náhodně vybranou blokovou šifru E útočník $A\_E\_E^{-1}$ pomocí algoritmu $A\_E\_E^{-1}(\sigma)$ získá hodnotu zprávy M, pro níž je $HMAC^E(M) = \sigma$, tj. nalezne vzor k dané haši. Označme $Adv\_inv\_HMAC[n](q) = Max \{Pr\_E\_\sigma\}$, kde maximum se bere přes všechny možné útočníky $A\_E\_E^{-1}(\sigma)$, používající dohromady nejvýše q volání orákula E nebo E^{-1}. Zvolme $n \in N$ a $K \geq n$. Potom pro libovolné $1 \leq q < 2^n$ platí

$$0.3 * q / 2^n \leq Adv\_inv\_HMAC[n](q) \leq 1.0 * q / 2^n.$$

3.3. Odolnost HMAC proti nalezení kolize

Věta 2: Odolnost HMAC proti nalezení kolize

Označme výhodu pro nalezení kolize jako reálné číslo

$Adv\_coll\_HMAC[n](A) = Pr[E \xleftarrow{\$} BC(K, n); (M, M') \xleftarrow{\$} A: M' \neq M \;\&\; HMAC^E(M) = HMAC^E(M')]$. Pro $1 \leq q$ definujeme $Adv\_coll\_HMAC[n](q) = Max \{Adv\_coll\_HMAC[n](A)\}$, kde maximum se bere přes všechny možné útočníky $A\_E\_E^{-1}$, používající dohromady nejvýše q volání orákula E nebo E^{-1}. Zvolme $n \in N$. Nechť $K \geq n \geq 3$. Potom pro libovolné $1 < q \leq 2^{n/2}$ platí

$$0.158 * q(q-2) / 2^n \leq Adv\_coll\_HMAC[n](q) \leq 1.5 * q(q-1) / 2^n.$$

3.4. Věty o bezpečnosti NMAC

Množinu všech náhodných orákul s p bitovým vstupem a q bitovým výstupem označujeme NO(p, q). NMAC definovanou výše pomocí orákul $f \in NO(K + n, n)$ a $g \in NO(n, n)$ označujeme jako $NMAC^{f,g}$, resp. krátce NMAC. Označením A_f_g značíme útočníka (jakýkoliv algoritmus útoku), který má přístup k orákulům f a g. $A\_f\_g(\sigma)$ označuje algoritmus, zvolený na základě parametru σ.

Konvence ([BRS02], str. 328). V případě NMAC budeme v dalším uvažovat následující významné předpoklady podobně jako u HMAC. Když útočník hledající kolizi NMAC jako výstup předkládá M a M´, pak se předpokládá, že musel projít výpočtem NMAC(M) a NMAC(M´) v tom smyslu, že se musel zeptat na všechny hodnoty f a g, které jsou použité ve všech iteracích během výpočtu NMAC(M) a NMAC(M´). Podobně, když útočník hledající vzor NMAC jako výstup předkládá zprávu M, předpokládáme, že musel projít výpočtem NMAC(M), tj. musel zjistit všechny hodnoty f a g, které jsou použité ve všech iteracích během výpočtu NMAC(M).

3.5. Odolnost NMAC proti nalezení vzoru

Věta 3: Odolnost NMAC proti nalezení vzoru

Nechť $Pr\_f\_g\_\sigma = Pr[f \xleftarrow{\$} NO(K + n, n); g \xleftarrow{\$} NO(n, n); \sigma \xleftarrow{\$} \{0,1\}^n; M \longleftarrow A\_f\_g(\sigma): NMAC(M) = \sigma]$ je pravděpodobnost jevu, že pro nějakou náhodně vybranou hodnotu haše σ a náhodně vybraná orákula f, g útočník $A\_f\_g(\sigma)$ získá hodnotu zprávy M, pro níž je $NMAC(M) = \sigma$, tj. nalezne vzor k dané haši. Označme $Adv\_inv\_NMAC[n](q) = Max \{Pr\_f\_g\_\sigma\}$, kde maximum se bere přes všechny možné algoritmy (útočníky) A_f_g,

používající dohromady nejvýše q volání orákula f nebo g. Zvolme n ∈ N. Potom pro libovolné $1 \le q < 2^n$ platí

$$0.3 * q/2^n \le \text{Adv\_inv\_NMAC[n]}(q) \le 1.0 * q/2^n.$$

3.6. Odolnost NMAC proti nalezení kolize

Věta 4. Odolnost NMAC proti nalezení kolize

Označme výhodu pro nalezení kolize jako reálné číslo Adv_coll_NMAC[n](A) = Pr[f $\xleftarrow{\$}$ NO(K + n, n); g $\xleftarrow{\$}$ NO(n, n); (M, M´) $\xleftarrow{\$}$ A: M´ ≠ M & NMAC(M) = NMAC(M´)]. Pro 1 ≤ q definujeme Adv_coll_NMAC[n](q) = Max { Adv_coll_NMAC[n](A) }, kde maximum se bere přes všechny možné algoritmy (útočníky) A_f_g, používající dohromady nejvýše q volání orákula f nebo g. Zvolme n ∈ N. Potom pro libovolné $1 < q \le 2^{n/2}$ platí

$$0.158 * q(q-2)/2^n \le \text{Adv\_coll\_NMAC[n]}(q) \le 0.5 * q(q-1)/2^n.$$

4. Nový koncept SBŠ a SNMAC

V této kapitole zavedeme pojem speciální blokové šifry a na její bázi definujeme hašovací funkci (SNMAC).

Připomeňme, co způsobilo problémy současných hašovacích funkcí třídy MD a SHA:

- blokové šifry, použité v kompresních funkcích, zpracovávají klíč a otevřený text zásadně odlišně, umožňují vzájemné řízení změn jednoho vstupu pomocí druhého,
- dílčí funkce umožňují propagaci diferencí ze vstupů na výstupy,
- dílčí funkce jsou slabě nelineární, existují vysoce pravděpodobné lineární vztahy mezi jejich vstupy a výstupy.

Biham [Bih05] navrhl, aby se v hašovacích funkcích začala používat technologie blokových šifer. Máme na mysli takové stavební bloky, které jsou silně nelineární a odolné proti lineární a diferenciální kryptoanalýze.

Uvažujme tedy, že v kompresní funkci f, $h_i = f(h_{i-1}, m_i)$ jakýmkoliv způsobem, třeba i několikrát, použijeme blokovou šifru.

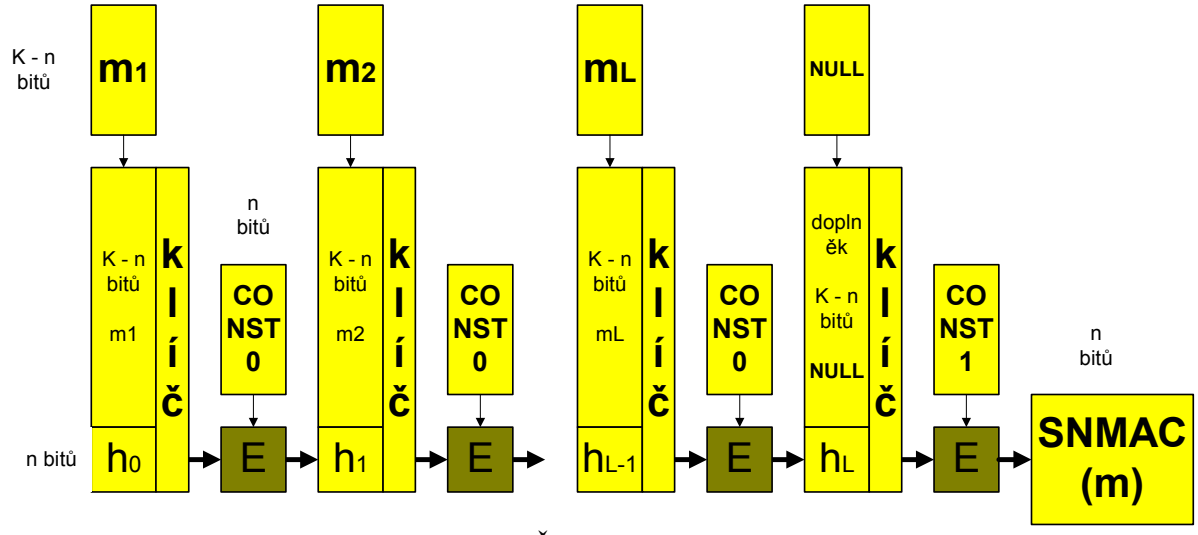
U současných útoků na hašovací funkce se v předpisu $h_i = f(h_{i-1}, m_i)$ vhodně mění současně h_{i-1} i m_i tak, aby vznikaly odpovídající diference v h_i. Protože funkci f realizuje nějaká bloková šifra a útočník může manipulovat všemi proměnnými, které do ní vstupují, může také manipulovat všemi proměnnými té blokové šifry. **U hašovacích funkcí tedy vzniká zvláštní situace, že útočník má možnost libovolně manipulovat otevřeným textem i klíčem použité blokové šifry**, a to nezávisle na tom, jakým způsobem je bloková šifra v hašovací funkci využita.

Způsobů použití blokových šifer pro konstrukce hašovacích funkcí byla studována celá řada. **Nikdy však nebyla klasická bloková šifra navrhována s předpokladem, že útočník bude mít možnost libovolně manipulovat s jejím klíčem.** Naopak, u většiny moderních šifer je klíč zpracováván slabšími funkcemi než datový vstup. Například u TripleDES je to lineární funkce, u AES slabě nelineární.

Homogenita. Aby nebylo možné využít ani slabin ve zpracování datového vstupu, ani ve zpracování klíče, požadujeme, aby u použité blokové šifry byly všechny proměnné bity

zpracovány stejně kvalitně a podobným způsobem. Tuto vlastnost nazýváme homogenitou. Homogenitu požadujeme také u výstupních bitů použité blokové šifry. Příkladem homogenně zpracovaných vstupních i výstupních bitů může být náhodný substituční box (permutace), i když bity výstupu jsou značně odlišné funkce vstupních bitů. Klasické blokové šifry téměř nikdy nesplňují požadavek homogenity. U většiny moderních šifer je klíčový vstup zpracováván slabšími funkcemi než datový vstup. Na druhé straně jsou téměř vždy homogenně zpracovávána množina bitů klíče a množina bitů otevřeného textu každá zvlášť. Proto požadovanou homogenitu můžeme docílit tak, že buď klíč nebo otevřený text volíme konstantní, zbylý vstup bude zpracován homogenně.

Speciální bloková šifra (SBŠ) a speciální NMAC (SNMAC). Uvažujme, že vlastnost homogenity u blokové šifry (E), použité v kompresní funkci (f), splníme tím, že všechny proměnné vstupy kompresní funkce (tj. datový blok m_i a kontext h_{i-1}) vedeme jako proměnnou $X = h_{i-1} \parallel m_i$ do otevřeného textu a klíč volíme konstantní: $f(X) = E_{Const0}(X)$. Kompresní funkce f by měla být jednocestná, aby bránila nalézání vzoru hašovací funkce, což tato konstrukce nesplňuje. Na druhou stranu blokové šifry byly vyvíjeny desítky let tak, aby ze znalosti šifrového a otevřeného textu nešel určit klíč, tj. zajišťují jednocestnost vzhledem ke klíči. Využijeme-li tohoto faktu, dostáváme konstrukci kompresní funkce přirozeně jako $f(X) = E_X(Const_0)$, tedy všechny proměnné bity vedou do klíče blokové šifry, a ta je použita pouze s konstantním otevřeným textem. Proto se dále budeme zabývat jen konstrukcí $f(X) = E_X(Const_0)$. V tomto případě E nazýváme speciální blokovou šifrou. Tento název si E určitě zaslouží, protože je použita pouze se dvěma různými konstantními otevřenými texty - $Const_0$ pro orákulum f a $Const_1$ pro orákulum g. Nyní můžeme definovat hašovací funkci SNMAC na bázi NMAC a SBŠ tak, jak ilustruje obr. 5.



Obr. 5: Definice SNMAC, založená na SBŠ a NMAC

Pojem SBŠ je nový a jeho definice se určitě bude ještě vyvíjet. Pro důkazy bezpečnosti konstrukce SNMAC budeme potřebujeme vlastnost, aby při pevném otevřeném textu ($Const_0$ a $Const_1$) byly $E: \{0, 1\}^K \times Const_0 \rightarrow \{0, 1\}^n : (k, Const_0) \rightarrow y = E_k(Const_0) = f(k)$ a $E: \{0, 1\}^K \times Const_1 \rightarrow \{0, 1\}^n : (k, Const_1) \rightarrow y = E_k(Const_1) = g(k)$ náhodná orákula vzhledem k proměnnému klíči. Aby f, g byly kvalitní funkce při libovolné volbě konstant $Const_0$ a $Const_1$, budeme požadovat, aby $E: \{0, 1\}^K \times \{0, 1\}^n \rightarrow \{0, 1\}^n : (k, x) \rightarrow y = E_k(x)$ byla kvalitní bloková šifra jakožto celé zobrazení s proměnným klíčem i proměnným otevřeným textem.

Všechny diferenční a lineární útoky, které jsou úspěšné u hašovacích funkcí, se u SBŠ převádí na diferenční a lineární útoky s využitím klíče. Proto na rozdíl od běžných blokových šifer bude od speciální blokové šifry požadováno, aby byla odolná proti různým diferenčním a lineárním útokům, vedeným zejména z klíčového vstupu. Požadavek můžeme rozšířit i na datový vstup (jako by byl proměnný) a na kombinaci datového a klíčového vstupu.

Požadujeme tedy, aby mezi proměnnými (k, x) a $y = E_k(x)$ neexistovaly žádné diferenční a lineární vztahy s využitelnou pravděpodobností. Jinými slovy, požadavky na SBŠ jsou stejné jako na klasickou blokovou šifru a navíc se požaduje silnější zpracování klíče. Klíč by měl být u SBC zpracováván se stejnou kryptografickou kvalitou Zpracování klíče by mělo být na stejné úrovni, jako je zpracováván otevřený text u klasických blokových šifer. Co tedy víme o SBŠ:
Speciální bloková šifra E:

- zpracovává klíč na stejné úrovni kvality jako datový vstup,
- zpracovává všechny bity klíče stejně kvalitně (homogenně),
- na rozdíl od klasických blokových šifer bude přirozené použít délku klíče obvykle mnohonásobně delší než délku bloku, například K = 4096, resp. 8192 a n = 256, resp. 512,
- je konstruována pomocí technologie blokových šifer,
- není primárně určena k šifrování dat,
- je použita v hašovací funkci s konstantním otevřeným textem, veškerá proměnná vstupuje do E prostřednictvím klíče,
- když uvažujeme, že má také proměnný otevřený text, měla by to být kryptograficky silná klasická bloková šifra,
- útočník může libovolně manipulovat s klíčem.

Definice SBŠ není uzavřena a musí se ještě dále zkoumat.

Definice. Hašovací funkce SNMAC. Hašovací funkce SNMAC je iterativní hašovací funkce typu NMAC ([BCK96], [CDMP05]), která využívá speciální blokovou šifru E s n bitovým blokem a K-bitovým klíčem. Má kompresní funkci f a závěrečnou úpravu g, kde
$f: \{0, 1\}^K \rightarrow \{0, 1\}^n : X \rightarrow E_X(Const_0)$,
$g: \{0, 1\}^n \rightarrow \{0, 1\}^n : X \rightarrow E_{X \,||\, NULL}(Const_1)$,
$K \geq n$, $Const_0$ a $Const_1$ jsou různé konstanty a NULL je řetězec K - n nulových bitů.

Hašování zprávy m má tři kroky.
Krok 1. Doplnění
Zprávu m, kterou hašujeme, nejprve doplníme bitem 1, nejmenším (i nulovým) počtem bitů 0 a 128bitovým číslem (které vyjadřuje délku m v bitech) tak, aby její délka byla L násobkem čísla K - n, kde L je přirozené číslo. Tuto doplněnou zprávu rozdělíme na L bloků o délce K - n bitů, $m = m_1 \,||\, \ldots \,||\, m_{L-1} \,||\, m_L$.
Definujeme (viz obr. 5) h_0 jako konstantu (inicializační hodnota)
Krok 2. Iterace
$h_i = f(h_{i-1} \,||\, m_i)$, i = 1, ..., L,
Krok 3. Závěrečná úprava
$SNMAC(m) = g(h_L)$.

Cíl útočníka. U klasických blokových šifer byl hlavním cílem útočníka klíč. U speciální blokové šifry může útočník s klíčem dokonce libovolně manipulovat. Vzniká otázka, co je nyní jeho cílem. Protože hašovací funkce SNMAC je založena na SBŠ, jeho cílem bude zejména nalézt vzor nebo kolizi SBŠ. Obecněji bude jeho cílem možnost jakýmkoliv způsobem řídit vztah mezi vstupem a výstupem SBŠ, což by mohlo vést k nalezení vlastností odlišujících hašovací funkci od náhodného orákula.

Blokovou šifru tak, jak ji dnes známe, budeme muset upravit. Libovolná manipulovatelnost s klíčem je pro současné blokové šifry naprosto nepřirozený požadavek, na který nejsou připraveny, a nemají proto k tomu vystavěna obranná opatření. Příprava klíče je většinou slabá, neporovnatelná se zpracováním otevřeného textu. Proto zpracování klíče musí být u SBŠ zesíleno na úroveň zpracování otevřeného textu v současných blokových šifrách.

Proč není vhodné použít kvalitní klasickou blokovou šifru pro konstrukci hašovací funkce? Z práce Corona a kol. [CDMP05] a důkazů z kapitoly 2 vyplývá, že konstrukce NMAC a HMAC jsou výpočetně bezpečné proti excesům typu kolize a nalezení vzoru. Nestačilo by proto například použít kvalitní klasickou blokovou šifru v konstrukci HMAC? Odpověď je záporná. Nevýhodou všech současných blokových šifer je, že zpracovávají klíč a datový vstup různým způsobem, nehomogenně a s různou kryptografickou silou. Tato nehomogenita byla využita k útokům na blokové šifry i k útokům na hašovací funkce (viz například [BDK03], [BDK05], [HKL05], [KBP05], [KHL04], [KKH04], [KLS04], [KML02], [SKH04], [Kli06a], [YWYP06] a nejnověji [BDK07]). Ze současných útoků na hašovací funkce vyplývá, že hodnoty vstupních dat a kontextu jsou stejně kryptograficky cenné, a proto by měly být zpracovány homogenně (stejně kvalitně). Tuto vlastnost nemá žádná současná bloková šifra. Stejně tak žádná současná bloková šifra nebyla konstruována s předpokladem, že útočník bude mít plnou kontrolu nad klíčem. Konstrukce SBŠ bude proto odlišná od klasických blokových šifer, i když může využívat jejich osvědčené stavební prvky.

5. Konkrétní instance SBŠ a SNMAC

Konstrukce SNMAC na bázi SBŠ je obecná a umožňuje využívat různé instance SBŠ. Jako konkrétní instanci SBŠ jsme navrhli algoritmus DN (Double Net) a s jeho využitím jsme obdrželi hašovací funkci HDN (Hash Double Net). Popisy DN a HDN jsou uvedeny v dodatcích 2 a 3. Jejich zdrojové kódy, testovací příklady apod. budou k dispozici po schválení jejich publikace na [Kli06b].

DN má délku klíče 8192 bitů a délku bloku 512 bitů. HDN má 512bitový kód a dosahuje rychlosti hašování 3 - 4x nižší než SHA-512.

Nižší rychlost hašování HDN oproti SHA-512 je pochopitelná po porovnání obou funkcí. SHA-512 používá slabší vnitřní nelineární funkce, zatímco HDN používá technologii blokových šifer a je bezpečnostně naddimenzovaná.

6. Závěr

Generické problémy hašovacích funkcí vyvolaly potřebu nového návrhu konceptu hašovacích funkcí. Nové důkazy bezpečnosti však umožňují využít konstrukci NMAC/HMAC, navrženou Bellare a kol. v roce 1996 [BCK96]. V tomto příspěvku poprvé dokazujeme kvantitativní odhady odolnosti těchto konstrukcí proti nalezení vzoru a kolize. Odtud vyplývá, že jsou výpočetně odolné i proti nalezení multivzorů a multikolizí. Coron a kol. [CDMP05] na CRYPTO 2005 ukázali, že NMAC/HMAC se limitně blíží k náhodnému orákulu. To spolu se zde předloženými kvantitativními důkazy dává velmi dobré záruky bezpečnosti těchto konstrukcí.

Druhá část příspěvku se zabývá praktickou konstrukcí funkcí NMAC/HMAC a návrhem hašovací funkce SNMAC na bázi blokové šifry. Ukazujeme, že blokové šifry by měly být použity v hašovacích funkcích jiným způsobem než dosud. Nazýváme je speciální blokové šifry (SBŠ) a formulujeme jejich vlastnosti. Toto nové kryptografické primitivum se vymyká klasické představě blokových šifer. Základní vlastností SBŠ je, že útočník má plnou kontrolu nad jejím klíčem. Proti takovému požadavku nebyly dosud blokové šifry konstruovány, a proto žádné současné blokové šifry nejsou příliš vhodné pro použití v hašovacích funkcích. Konkrétní příklad SBŠ je uveden v [Kli06b].

Navrhujeme novou třídu hašovacích funkcí SNMAC jako konstrukci typu NMAC s využitím speciální blokové šifry. Tento koncept může být kandidátem na hašovací funkce nové generace. Má dokazatelnou výpočetní odolnost proti nalezení vzoru a kolize, limitně se blíží náhodnému orákulu a umožňuje návrh různých instancí pomocí různých SBŠ.

Jako příklad také navrhujeme speciální blokovou šifru DN (Double Net) a definujeme hašovací funkci HDN (Hash Double Net) jako konstrukci SNMAC na bázi DN.

Poděkování. Autor děkuje Tomáši Rosovi za mnoho užitečných připomínek k předchozím verzím příspěvku.

7. Literatura

[BCK96] M. Bellare, R. Canetti and H. Krawczyk. Keying hash functions for message authentication. Advances in Cryptology – CRYPTO ’96, Lecture Notes in Computer Science Vol. 1109, pp. 1-15, Springer-Verlag, 1996.

[Bel06] M. Bellare. New Proofs for NMAC and HMAC: Security without Collision-Resistance. To be published, Advances in Cryptology – CRYPTO ’06, Lecture Notes in Computer Science Vol. 4117, Springer-Verlag, 2006, Cryptology ePrint Archive, Report 2006/043.

[BCJ05] E. Biham, R. Chen, A. Joux, P. Carribault, Ch. Lemuet and W. Jalby. Collisions of SHA-0 and Reduced SHA-1. Advances in Cryptology –EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 36–57, Springer-Verlag, 2005.

[BDK03] E. Biham, O. Dunkelman, and N. Keller. Rectangle Attacks on 49-Round SHACAL-1, FSE 2003, Lecture Notes in Computer Science Vol. 2887, pp. 22-35, Springer-Verlag, 2003.

[BDK05] E. Biham, O. Dunkelman, and N. Keller. Related-Key Boomerang and Rectangle Attacks, Advances in Cryptology – EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 507–525, Springer-Verlag, 2005.

[BDK07] E. Biham, O. Dunkelman, and N. Keller. A Simple Related-Key Attack on the Full SHACAL-1, to be published, CT-RSA 2007, RSA Conference 2007, Cryptographers' Track, February 5-9, 2007, Moscone Center, San Francisco, USA.

[Bih05] E. Biham: Recent advances in hash functions and the way to go, Conference on Hash Functions (Ecrypt Network of Excellence in Cryptology), June 23-24, 2005, Przegorzaly (Krakow), Poland, http://www.ecrypt.eu.org/stvl/hfw/Biham.ps.

[BRS02] J. Black, P. Rogaway, T. Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. Advances in Cryptology – CRYPTO 2002, Lecture Notes in Computer Science Vol. 2442, pp. 320-335, Springer-Verlag, 2002. Extended version: Cryptology ePrint Archive, Report 2002/066, http://eprint.iacr.org/2002/066.

[CDMP05] J. S. Coron, Y. Dodis, C. Malinaud and P. Puniya. Merkle-Damgard Revisited: how to construct a hash-function. Advances in Cryptology – CRYPTO 2005, Lecture Notes in Computer Science Vol. 3621, pp. 430 - 448, Springer-Verlag, 2005.

[Dam89] I. Damgard. A Design Principle for Hash Functions. Advances in Cryptology - CRYPTO 1989, Lecture Notes in Computer Science Vol. 435, pp. 416–427, Springer-Verlag, 1990.

[EM91] S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. In Advances in Cryptology – ASIACRYPT '91, Lecture Notes in Computer Science Vol. 739, pp. 210–224. Springer-Verlag, 1992.

[GHA06] P. Gauravaram, S. Hirose and S. Annadurai. An Update on the analysis and design of NMAC and HMAC functions. To be published in International Journal of Network Security

[HPR04] P. Hawkes, M. Paddon, and G. G. Rose. On Corrective Patterns for the SHA-2 Family. Cryptology ePrint Archive, Report 2004/207, 2004.

[HKK03] S. Hong, J. Kim, G. Kim, J. Sung, C. Lee and S. Lee. Impossible Differential Attack on 30-Round SHACAL-2, INDOCRYT 2003, Lecture Notes in Computer Science Vol. 2904, pp. 97-106, Springer-Verlag, 2003.

[HKL05] S. Hong, J. Kim, S. Lee and B. Preneel. Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192, FSE 2005, Lecture Notes in Computer Science Vol. 3557, pp. 368–383, Springer-Verlag, 2005.

[Jou04] A. Joux. Multicollisions in Iterated Hash Functions. Advances in Cryptology - CRYPTO 2004, Lecture Notes in Computer Science Vol. 3152, pp. 306–316, Springer-Verlag, 2004.

[KML02] J. Kim, D. Moon, W. Lee, S. Hong, S. Lee, and S. Jung. Amplified Boomerang Attack against Reduced-Round SHACAL, Advances in Cryptology - ASIACRYPT 2002, Lecture Notes in Computer Science Vol. 2501, pp. 243 - 253, Springer-Verlag, 2002.

[KBP05] J. Kim, A. Biryukov, B. Preneel, and S. Lee. On the Security of Encryption Modes of MD4, MD5 and HAVAL, ICICS 2005, Lecture Notes in Computer Science Vol. 3783, pp. 147-158, Springer-Verlag, 2005.

[KK05] J. Kelsey and T. Kohno.Herding Hash Functions and the Nostradamus Attack, Cryptographic Hash Workshop, held in NIST, Gaithersburg, Maryland, 2005, IACR Cryptology ePrint Archive, Report 2005/281, 2005.

[KKH04] J. Kim, G. Kim, S. Hong, S. Lee and D. Hong.The Related-Key Rectangle Attack-Application to SHACAL-1, ACISP 2004, Lecture Notes in Computer Science Vol. 3108, pp. 123-136, Springer-Verlag, 2004.

[KKL04] J. Kim, G. Kim, S. Lee, J. Lim and J. Song. Related-Key Attacks on Reduced Rounds of SHACAL-2, INDOCRYPT 2004, Lecture Notes in Computer Science Vol. 3348, pp. 36 - 44, Springer-Verlag, 2004.

[Kli06a] V. Klima. Tunnels in Hash Functions: MD5 Collisions Within a Minute, Cryptology ePrint Archive, Report 2006/105, 18 March, 2006.

[Kli06b] SNMAC homepage http://cryptography.hyperlink.cz/SNMAC/SNMAC.html

[KLS04] J. Kim, G. Kim, S. Lee, J. Lim and J. Song. Related-Key Attacks on Reduced Rounds of SHACAL-2, INDOCRYPT 2004, Lecture Notes in Computer Science Vol. 3348, pp. 36 - 44, Springer-Verlag, 2004.

[KML02] J. Kim, D. Moon, W. Lee, S. Hong, S. Lee, and S. Jung. Amplified Boomerang Attack against Reduced-Round SHACAL, Advances in Cryptology - ASIACRYPT 2002, Lecture Notes in Computer Science Vol. 2501, pp. 243 - 253, Springer-Verlag, 2002.

[KR96] J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. Journal of Cryptology, 14(1):17–35, 2001. Earlier version in CRYPTO '96.

[KS05] J. Kelsey and B. Schneier. Second Preimages on n-Bit Hash Functions for Much Less than 2^n. Advances in Cryptology - EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 474–490, Springer-Verlag, 2005.

[MD5] R. Rivest. The MD5 message-digest algorithm, Internet RFC 1321, April 1992.

[Mer89] R. C. Merkle. One Way Hash Functions and DES. Advances in Cryptology - CRYPTO 1989, Lecture Notes in Computer Science Vol. 435, pp. 428–446, Springer-Verlag, 1990.

[MMO85] S. M. Matyas, C. H. Meyer and J. Oseas. Generating strong one-way functions with cryptographic algorithm. IBM Techn. Disclosure Bull., Vol. 27, No. 10A, 1985, pp. 5658 - 5659.

[MPRR06a] F. Mendel, N.Pramstaller, C.Rechberger, and V.Rijmen. Analysis of Step-Reduced SHA-256, to be published, FSE 2006

[MPRR06b] F.Mendel, N.Pramstaller, C.Rechberger, and V.Rijmen. The Impact of Carries on the Complexity of Collision Attacks on SHA-1, to be published, FSE 2006

[S49] C. Shannon. Communication theory of secrecy systems. Bell Systems Technical Journal, 28(4):656–715, 1949.

[Sch04] B. Schneier. Cryptanalysis of MD5 and SHA. Crypto-Gram Newsletter, September 2004, http://www.schneier.com/crypto-gram-0409.html#3

[SHA-0] National Institute of Standards and Technology. Secure hash standard. Federal Information Processing Standard, FIPS PUB 180, May 1993.

[SHA-1] National Institute of Standards and Technology. Secure hash standard. Federal Information Processing Standard, FIPS PUB 180-1, April 1995.

[SHA-2] National Institute of Standards and Technology. Secure hash standard. Federal Information Processing Standard, FIPS PUB 180-2, August 2000.

[SKH04] Y. Shin, J. Kim, G. Kim, S. Hong and S. Lee. Differential-Linear Type Attacks on Reduced Rounds of SHACAL-2, ACISP 2004, Lecture Notes in Computer Science Vol. 3108, pp. 110–122. , Springer-Verlag, 2004.

[Tsu92] G. Tsudik. Message authentication with one-way hash functions. ACM Computer Communications Review, 22(5):29-38, 1992.

[W84] R. Winternitz. A secure one-way hash function built from DES. In Proceedings of the IEEE Symposium on Information Security and Privacy, pp. 88–90. IEEE Press, 1984.

[WY05] X. Wang and H. Yu. How to Break MD5 and Other Hash Functions. Advances in Cryptology - EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 19–35, Springer-Verlag, 2005.

[WYY05a] X. Wang, H. Yu and Y. L. Yin. Efficient Collision Search Attacks on SHA-0. Advances in Cryptology - CRYPTO ’05, Lecture Notes in Computer Science Vol. 3621, pp. 1–16, Springer-Verlag, 2005.

[WYY05b] X. Wang, Y. L. Yin and H. Yu. Finding collisions in the full SHA-1. Advances in Cryptology - CRYPTO ’05, Lecture Notes in Computer Science Vol. 3621, pp. 17–36, Springer-Verlag, 2005.

[YB05] H. Yoshida and A. Biryukov. Analysis of a SHA-256 Variant, SAC 2005, Lecture Notes in Computer Science Vol. 3897, pp. 245 – 260, Springer-Verlag, 2005.

[YBP05] H. Yoshida, A. Biryukov, and B. Preneel. Some applications of the Biham-Chen attack to SHA-like hash functions, CRYPTOGRAPHIC HASH WORKSHOP, NIST, Gaithersburg, Maryland, USA, October 31 - November 1, 2005, http://csrc.nist.gov/pki/HashWorkshop/2005/Oct31_Presentations/Yoshida_cameraNistHash.pdf

[YWYP06] H.Yu, X.Wang, A.Yun and S. Park. Cryptanalysis of the Full HAVAL with 4 and 5 Passes. To be published, FSE 2006.

8. Dodatek 1: Důkazy vět

V důkazech budeme používat následující lemma.

Lemma 1.

(1) pro každé $x \in \langle 0, 1 \rangle$ platí $1 - e^{-x} \geq (1 - e^{-1})\, x$,

(2) pro každé $x \in \langle 0, 1 \rangle$ platí $e^{-x} \geq 1 - x$,

(3) pro každé $x \in (0, 1)$ a přirozené q platí $1 - qx \leq (1 - x)^q$,

(4) pro každé $x \in (0, 1)$ a přirozené q platí $(1 - x/q)^q \leq e^{-x}$.

Důkaz. Důkaz vyplývá z vlastností mocninné a exponenciální funkce.

8.1. Důkaz Věty 1

Nyní dokazujeme $0.3*q/2^n \le \text{Adv\_inv\_HMAC}[n](q)$.

Nechť E je náhodně vybraná bloková šifra z množiny BC(K, n) a σ náhodně vybraná hodnota z množiny R = $\{0, 1\}^n$. K důkazu věty postačí ukázat, že Pr_E_σ ≥ 0.3 * $q/2^n$ pro námi zvoleného útočníka A. Definujme útočníka. Útočník A hledá jeden K-bitový blok m, pro nějž $h_1 = E_m(h_0) \oplus h_0$ a $E_{h1\_pad}(IV) = \sigma$. Přitom se dotazuje orákula E celkem q (=2* q/2) krát, dotazy na E^{-1} nevyužívá. Pro každé $1 \le i \le q/2$ libovolným způsobem volí klíč k_i a od orákula E obdrží hodnotu $E_{ki}(h_0)$, vypočte $y_i = E_{ki}(h_0) \oplus h_0$ a od orákula obdrží hodnotu $H_i = HMAC^E(k_i) = E_{yi\_pad}(IV)$. Pokud $H_i = \sigma$ pro nějaké i, útočník A našel vzor hašovací hodnoty σ a vrátí jako hledanou zprávu $M = k_i$. Pokud $E_{yi\_pad}(IV)$ není rovno σ pro žádné $1 \le i \le q/2$, vrátí negativní odpověď. Z definice náhodné blokové šifry vyplývá, že když fixujeme otevřený text x (zde $x = h_0$ a x = IV), pak zobrazení $\{0,1\}^K \rightarrow \{0,1\}^n : k \rightarrow E_k(x)$ je náhodným orákulem s K-bitovým vstupem a n-bitovým výstupem. Proto $\{y_i\}_{1 \le i \le q}$ obsahuje q/2 různých náhodných hodnot z množiny $\{0, 1\}^n$ a $\{y_i\_pad\}_{1 \le i \le q/2}$ je množina q/2 různých hodnot z množiny$\{0, 1\}^K$. Odtud ze stejného důvodu (x = IV) vyplývá, že $\{H_i\}_{1 \le i \le q/2}$ je množina q/2 náhodně vybraných hodnot z množiny $\{0, 1\}^n$. Pravděpodobnost P, že se v množině $\{H_i\}_{1 \le i \le q/2}$ vyskytne hodnota σ je

$$P = 1 - \left(1 - \frac{1}{2^n}\right)^{q/2} \ge^{(4)} 1 - e^{-q/2^{n+1}} \ge^{(1)} (1 - e^{-1}) * q / 2^{n+1} \ge 0.3 * q / 2^n$$

s využitím Lemmatu 1 (4), (1) a skutečnosti, že $q/2^{n+1}<1$, cbd.

Nyní dokazujeme $\text{Adv\_inv\_HMAC}[n](q) \le 1.0*q/2^n$.

Nechť E je náhodně vybraná bloková šifra z množiny BC(K, n) a σ náhodně vybraná hodnota z množiny R = $\{0, 1\}^n$. Nechť A je nějaký útočník. K důkazu věty postačí ukázat, že Pr_E_σ ≤ $q/2^n$. Ať má útočník A jakoukoliv strategii, může využívat pouze dotazů na orákulum E nebo E^{-1}. Během činnosti algoritmu A si orákulum E vytváří seznam záznamů o dotazech a odpovědích (k_i, x_i, y_i) a orákulum E^{-1} seznam (k_j, x_j, y_j), kde vstupem E je (k_i , x_i) a výstupem $y_i = E_{ki}(x_i)$; vstupem E^{-1} je (k_j, y_j) a výstupem $x_j = E^{-1}_{kj}(y_j)$. Rozlišujeme ještě případy, kdy x_i = IV, $x_i \ne$ IV, $y_j = \sigma$, $y_j \ne \sigma$. Celkem máme

q_1 dotazů na E typu (k_i, IV, y_i),
q_3 dotazů na E typu (k_i, ≠IV, y_i),
q_2 dotazů na E^{-1} typu (k_j, x_j, σ),
q_4 dotazů na E^{-1} typu (k_j, x_j, ≠σ),
kde $q_1 + q_2 + q_3 + q_4 = q$.

Jestliže je algoritmus útočníka A úspěšný, pak se v seznamu (q_1) musí objevit alespoň jeden záznam, kde $y_i = \sigma$ nebo v seznamu (q_2) alespoň jeden záznam, kde x_j = IV, což odpovídá závěrečné operaci hašovací funkce. Dotazy v seznamech (q_3) a (q_4) není nutné zkoumat. Z definice náhodné blokové šifry vyplývá, že když fixujeme otevřený text x (zde x = IV), pak zobrazení $\{0,1\}^K \rightarrow \{0,1\}^n : K \rightarrow E_K(x)$ je náhodným orákulem s K-bitovým vstupem a n-bitovým výstupem. Proto množina $\{y_i\}_{1 \le i \le q1}$ ze seznamu (q_1) je množina q_1 náhodných hodnot z množiny $\{0, 1\}^n$. Z definice náhodné blokové šifry vyplývá, že když fixujeme šifrový text y (zde y = σ pro libovolné σ), pak zobrazení $\{0,1\}^K \rightarrow \{0,1\}^n : k \rightarrow E^{-1}_k(\sigma)$ je náhodným orákulem s K-bitovým vstupem a n-bitovým výstupem. Proto množina $\{x_j\}_{1 \le j \le q2}$

ze seznamu (q_2) je množina q_2 náhodně vybraných hodnot z množiny $\{0, 1\}^n$. Pravděpodobnost P, že se v množině $\{y_i\}_{1 \le i \le q1}$ vyskytne hodnota σ nebo v množině $\{x_j\}_{1 \le j \le q2}$ vyskytne hodnota IV, je

$$P = 1 - \left(1 - \frac{1}{2^n}\right)^{q_1} * \left(1 - \frac{1}{2^n}\right)^{q_2} \le 1 - \left(1 - \frac{1}{2^n}\right)^{q} \le^{(3)} q / 2^n.$$

Využili jsme přitom Lemmatu 1 (3). Tím je důkaz Věty 1 ukončen.

8.2. Důkaz Věty 2

Nyní budeme dokazovat $0.158 * q(q-2)/2^n \le \text{Adv\_coll\_HMAC[n](q)}$

K důkazu věty postačí ukázat $\text{Adv\_coll\_H[n]}(A) \ge 0.158 * q(q-2)/2^n$ pro námi zvoleného útočníka A. Definujme útočníka. Útočník A hledá kolidující zprávy $x_i \ne x_j$, které včetně doplnění mají jeden blok o K bitech:

$y_i = E_{xi}(h_0)$ a $E_{(yi \oplus h0)\_pad}(IV) = \sigma$ a

$y_j = E_{xj}(h_0)$ a $E_{(yj \oplus h0)\_pad}(IV) = \sigma$ pro nějaké σ.

Poznamenejme, že kolize může nastat

(1) po zašifrování h_0 v hodnotách y_i nebo

(2) až v druhém kroku, po zašifrování IV různými klíči $(y_i \oplus h_0)\_pad$ a $(y_j \oplus h_0)\_pad$.

Těmto dvěma možnostem odpovídá postup útočníka. Útočník se dotazuje orákula E celkem q krát. Činí q_1 dotazů na zašifrování hodnoty h_0 a q_2 dotazů na zašifrování hodnoty IV, $q_1 = q_2 = q/2$. Pro každé $1 \le i \le q_1$ útočník A libovolným způsobem volí K bitové klíče k_i a od orákula E obdrží hodnoty $y_i = E_{ki}(h_0)$. Vytvoří seznam (k_i, h_0, y_i). Pokud v něm pro různá i a j najde dvě shodné hodnoty y_i a y_j, obdrží kolizi pro zprávy k_i a k_j. Pokud ne, je v tomto seznamu q_1 různých náhodných n-bitových hodnot v první položce a útočník vytváří druhý seznam. Pro každé $1 \le i \le q_2$ vybírá z předchozího seznamu hodnoty y_i a od orákula E obdrží hodnoty $Y_i = E_{(yi \oplus h0)\_pad}(IV)$. Vytvoří seznam $((y_i \oplus h_0)\_pad, IV, Y_i)_{1 \le i \le q2}$. Pokud v něm najde dvě stejné n bitové hodnoty Y_i a Y_j pro různé i a j, obdrží kolizi pro zprávy k_i a k_j. Pokud A kolizi nenajde, vrací negativní odpověď (kolize nenalezena). Označme p pravděpodobnost, že A kolizi tímto postupem nalezne a P, že nenalezne. Zde využijeme faktu, že z definice náhodné blokové šifry vyplývá, že když fixujeme otevřený text x (zde použijeme tento fakt pro dvě hodnoty, a to $x = h_0$ a $x = IV$), pak zobrazení $\{0,1\}^K \rightarrow \{0,1\}^n : k \rightarrow E_k(x)$ je náhodným orákulem s K-bitovým vstupem a n-bitovým výstupem. Proto $\{y_i\}_{1 \le i \le q1}$ je množina q_1 náhodných hodnot z množiny $\{0, 1\}^n$ a $\{ (yi \oplus h0)\_pad\}\}_{1 \le i \le q1}$ množina q_1 hodnot z množiny $\{0, 1\}^K$. Ze stejného důvodu je $\{Y_i\}_{1 \le i \le q2}$ množina q_2 náhodně vybraných hodnot z množiny $\{0, 1\}^n$. Pravděpodobnost, že ani v jednom z q_1 kroků (1) a ani v jednom z q_2 kroků (2) nenalezneme kolizi, je

$$P = \prod_{i=1}^{q1-1}\left(1 - \frac{i}{2^n}\right) * \prod_{i=1}^{q2-1}\left(1 - \frac{i}{2^n}\right) \le^{(2)} \prod_{i=1}^{q1-1}\left(e^{-i/2^n}\right) * \prod_{i=1}^{q2-1}\left(e^{-i/2^n}\right) =$$

$$= e^{-q1(q1-1)/2^{n+1} - q2(q2-1)/2^{n+1}} = e^{(-q(q-2)/4)/2^n}.$$

V úpravě jsme využili faktu (2) z Lemmatu 1 a volby $q_1 = q_2 = q/2$. Pravděpodobnost, že kolize nastane je

$$p = 1 - P \ge 1 - e^{(-q(q-2)/4)/2^n} \ge^{(1)} (1 - e^{-1}) * (q(q-2)/4)/2^n$$

$$\ge 0.158 * q(q-2)/2^n.$$

V úpravě jsme využili faktu (1) z Lemmatu 1, tj. že $1 - e^{-x} \geq (1-e^{-1})x$ pro všechna $0 \leq x \leq 1$. V roli x vystupuje výraz $x = (q(q-2)/4) / 2^n$. Protože podle předpokladů věty je vždy $q \leq 2^{n/2}$, máme $x \leq 1/4$, takže $x \leq 1$ a úprava byla korektní. Tím je důkaz ukončen.

Nyní budeme dokazovat $\mathrm{Adv\_coll\_HMAC[n](q)} \leq 1.5 * q(q-1)/2^n$.

Musíme ukázat platnost odhadu $\mathrm{Adv\_coll\_HMAC[n](A)} \leq 1.5 * q(q-1)/2^n$ pro libovolného útočníka A. Činnost útočníka si můžeme modelovat pomocí jeho dotazů orákulům E a E^{-1}. Vytváříme tabulku T záznamů (x, h, y, h ⊕ y), kde útočník volí klíč x a jednu z hodnot y nebo h. Obdrží $y = E_x(h)$ v případě dotazu orákulu E nebo $h = E^{-1}{}_x(y)$ v případě dotazu orákulu E^{-1}. Čtvrtá položka (h ⊕ y) je navíc a je informativní. Tabulka T může mít nejvíce q záznamů a slouží orákulům k tomu, aby na již dotazované hodnoty odpověděly tak jako v předchozím případě. Tabulka záznamů je jednotná pro obě orákula E a E^{-1}, takže pokud orákulum E na dotaz (x, h) odpovědělo y, orákulum E^{-1} na dotaz (x, y) odpoví h. Protože orákulum pořizuje záznamy jen pro nové dotazy, v tabulce nemohou být žádné dva záznamy, které mají stejné souřadnice (x, h) nebo (x, y). Útok končí nezdarem, pokud je tabulka naplněna q záznamy. Útočník A hledá kolidující zprávy $M \neq M'$, které včetně doplnění mají jeden až několik K-bitových bloků. Zprávy mohou mít odlišný počet bloků. Pokud nastane kolize, jsou dvě možnosti:

- **Vnitřní kolize.** Kolize nastala u obou zpráv před závěrečnou úpravou hašovací funkce. Útočník může v takovém případě příslušné komprese ukončit a přejít k závěrečné úpravě.
- **Závěrečná kolize.** Kolize nenastala nikdy před závěrečnou úpravou a nastala až v závěrečné úpravě hašovací funkce.

Vnitřní kolize

Uvažujme nejprve vnitřní kolizi. V tomto případě musí při hašování první zprávy v tabulce T vzniknout záznam $(x_i, h_{i-1}, y_i, h_{i-1} \oplus y_i)$ a při hašování druhé zprávy záznam $(x_j, h_{j-1}, y_j, h_{j-1} \oplus y_j)$, kde $i \neq j$ jsou čísla záznamů (nejsou to indexy bloků zpráv), $(x_i, h_{i-1}) \neq (x_j, h_{j-1})$ a současně $h_{i-1} \oplus y_i = h_{j-1} \oplus y_j$. Pro pevné $i = 2, ..., q$ označme C_i jev, že existuje j, $1 \leq j < i$ tak, že j-tý záznam v tabulce a i-tý záznam v tabulce mají stejné hodnoty ve čtvrté položce. Pravděpodobnost jevu C_i označme $\Pr[C_i]$. Označme t počet těch záznamů j v tabulce T, $1 \leq j < i$, pro něž je první položka x_j rovna x_i a současně $h_{j-1} \neq h_{i-1}$. Máme $0 \leq t \leq i - 1$. U těchto položek máme stejný klíč (x_i) a t různých hodnot h v druhé položce. Pro daný klíč x_i a pro t různých hodnot h tedy již máme zaznamenaných t různých hodnot $E_{xi}(h)$ ve třetí položce. Proto odpověď $y_i = E_{xi}(h_i)$ na i-tý dotaz (h_i) vybírá orákulum náhodně z množiny o $2^n - t$ hodnotách. Protože h_i je konstanta, výraz $y_i \oplus h_i$ je také vybírán náhodně z množiny o $2^n - t$ hodnotách. Pravděpodobnost, že se rovná některé z nejvýše (i - 1) čtvrtých položek tabulky T je menší nebo roven $(i - 1)/ (2^n - t)$. Proto platí

$$\Pr[C_i] \leq (i-1)/(2^n - t) \leq (i-1)/(2^n - (i-1)).$$

Zde jsme využili faktu, že z definice náhodné blokové šifry vyplývá, že pro každé pevné $k \in \{0,1\}^K$ zobrazení $\{0,1\}^n \rightarrow \{0,1\}^n : x \rightarrow E_k(x)$ je náhodnou permutací $\{0,1\}^n$. Protože $i \leq q \leq 2^{n/2} \leq 2^{n-1} + 1$, máme

$$\Pr[C_i] \leq (i-1)/(2^n - (i-1)) \leq (i-1)/2^{n-1}.$$

Pravděpodobnost, že v seznamu nastane vnitřní kolize, označme P_{int}. Máme

$$P_{\mathrm{int}} = \Pr[C_2] + \Pr[C_3] + \ldots + \Pr[C_q] \leq \sum_{i=2}^{q} \frac{i-1}{2^{n-1}} = q(q-1)/2^n.$$

Závěrečná kolize

Nyní budeme uvažovat, že nenastala vnitřní kolize a nastala závěrečná kolize. Označme P_{fin} pravděpodobnost tohoto jevu. Pro jednoduchost uvažujme, že v tabulce T jsou pouze záznamy, které mají jako druhou položku hodnotu IV (eventuelně o tento počet snížíme q). Protože uvažujeme pouze K ≥ n, jako klíč pro závěrečnou úpravu se bere předchozí n bitová hodnota typu h_N, která se eventuelně doplňuje nulovými bity na K bitový klíč. Závěrečná kolize znamená, že v tabulce T existují alespoň dva záznamy s různými klíči $x_i \neq x_j$ v první položce, stejnou hodnotou IV v druhé položce a stejnou hodnotou ve třetí položce $E_{xi}(IV) = E_{xj}(IV)$. Pro i = 2, ..., q označme C_i jev, že existuje $1 \le j < i$ tak, že j-tý záznam v tabulce a i-tý záznam v tabulce mají stejné hodnoty ve třetí položce, tj. $E_{xi}(IV) = E_{xj}(IV)$. Pravděpodobnost jevu C_i označme $\Pr[C_i]$. Protože zobrazení $\{0, 1\}^K \to \{0, 1\}^n : k \to E_k(IV)$ je náhodné orákulum, hodnota $E_{xi}(IV)$ je vybírána náhodně z množiny všech 2^n hodnot, a proto

$\Pr[C_i] \le (i-1)/2^n$. Jestliže existuje závěrečná kolize, pak musel nastat některý z jevů C_i pro $2 \le i \le q$. Odtud máme

$$P_{fin} \le \Pr[C_2] + \Pr[C_3] + \ldots + \Pr[C_q] \le \sum_{i=2}^{q} \frac{i-1}{2^n} \le q(q-1)/2^{n+1}.$$

Celkový odhad

Jestliže útočník nalezne kolizi, musí nastat alespoň jeden z předchozích případů. Odtud dostáváme

$$\mathrm{Adv\_coll\_HMAC[n](A)} \le$$

$$\le P_{\mathrm{int}} + P_{fin} \le q(q-1)/2^n + q(q-1)/2^{n+1} = 1.5 * q(q-1)/2^n,$$

což jsme měli dokázat.

8.3. Důkaz Věty 3

Nyní budeme dokazovat $0.3 * q/2^n \le \mathrm{Adv\_inv\_NMAC[n]\,(q)}$.

K důkazu věty postačí ukázat, že $\Pr\_f\_g\_\sigma \ge 0.3 * q/2^n$ pro námi zvoleného útočníka A. Definujme útočníka. Útočník A hledá jeden K-bitový blok m, pro nějž $h_1 = f(m, h_0)$ a $g(h_1) = \sigma$. Přitom se dotazuje orákula f celkem q_1 krát a orákula g q_2 krát, kde $q_1 = q_2 = q/2$. Pro každé $1 \le i \le q_1$ libovolným způsobem volí k_i a od orákula f obdrží hodnotu $y_i = f(k_i, h_0)$. Dále vypočte $H_i = g(y_i)$. Pokud $H_i = \sigma$ pro nějaké i, útočník A našel vzor hašovací hodnoty σ a vrátí jako hledanou zprávu $M = k_i$. Pokud $H_i \neq \sigma$ pro žádné $1 \le i \le q_1$, vrátí negativní odpověď. Protože f je náhodným orákulem, je $\{y_i\}_{1 \le i \le q1}$ množina q_1 náhodných hodnot z množiny $\{0, 1\}^n$. Protože g je náhodným orákulem, $\{H_i\}_{1 \le i \le q2}$ je množina q_2 náhodně vybraných hodnot z množiny $\{0, 1\}^n$. Pravděpodobnost P, že se v množině $\{H_i\}_{1 \le i \le q2}$ vyskytne hodnota σ je

$$P = 1 - \left(1 - \frac{1}{2^n}\right)^{q_2} \ge^{(4)} 1 - e^{-q_2/2^n} \ge^{(1)} (1 - e^{-1}) * q_2/2^n \ge$$

$$\ge 0.6 * q_2/2^n = 0.3 * q/2^n$$

s využitím Lemmatu 1 (4), (1) a skutečnosti, že $q/2^n < 1$, cbd.

Nyní budeme dokazovat $\mathrm{Adv\_inv\_NMAC[n](q)} \le q/2^n$.

Nechť A je nějaký útočník (algoritmus) a σ nějaká hodnota haše. K důkazu věty postačí ukázat, že Pr_f_g_σ ≤ q/2^n. Během činnosti algoritmu A si orákulum f vytvoří seznam q_1 záznamů (x_i, h_i, z_i), kde $z_i = f(x_i, h_i')$, a orákulum g seznam q_2 záznamů (h_j, y_j), kde $y_j = g(h_j)$, kde $q_1 + q_2 = q$. Jestliže je algoritmus útočníka A úspěšný, pak se v druhém seznamu musí objevit alespoň jeden záznam, kde $y_j = \sigma$. Z definice náhodného orákula g vyplývá, že množina $\{y_j\}_{1 \le j \le q2}$ z druhého seznamu je množina q_2 náhodně vybraných hodnot z množiny $\{0, 1\}^n$. Pravděpodobnost P, že se v množině $\{y_j\}_{1 \le j \le q2}$ vyskytne hodnota σ je

$$P = 1 - \left(1 - \frac{1}{2^n}\right)^{q_2} \le^{(3)} q_2 / 2^n \le q / 2^n.$$

Využili jsme přitom Lemmatu 1 (3). Tím je důkaz ukončen.

8.4. Důkaz Věty 4

Nyní budeme dokazovat

$$\mathrm{Adv\_coll\_NMAC[n](q)} \ge 0.158 * q(q-2)/2^n.$$

K důkazu věty postačí ukázat

$$\mathrm{Adv\_coll\_NMAC[n](A)} \ge 0.158 * q(q-2)/2^n.$$

pro námi zvoleného útočníka A. Definujme útočníka. Útočník A hledá kolidující zprávy $x_i \neq x_j$, které včetně doplnění mají jeden blok o K bitech. Platí tedy

$y_i = f(x_i, h_0)$ a $g(y_i) = \sigma$ a

$y_j = f(x_j, h_0)$ a $g(y_j) = \sigma$

pro nějaké σ.

Poznamenejme, že kolize může nastat

- v prvním kroku ($y_i = y_j$) nebo
- až v druhém kroku ($y_i \neq y_j$), po závěrečném zpracování orákulem g

Těmto dvěma možnostem odpovídá postup útočníka. Útočník se dotazuje orákula f celkem q_1 krát a orákula g celkem q_2 krát, kde $q_1 = q_2 = q/2$. Pro každé $1 \le i \le q_1$ útočník A libovolným způsobem volí K bitové vstupy x_i a vytvoří seznam (x_i, h_0, y_i), kde $y_i = f(x_i, h_0)$. Pokud v něm pro různá i a j najde dvě hodnoty y_i a y_j, které jsou stejné, obdrží kolizi pro zprávy x_i a x_j. Pokud kolizi nenajde, vytváří druhý seznam. Pro každé $1 \le i \le q_2$ volí y_i z předchozího seznamu a od orákula g obdrží $Y_i = g(y_i)$. Vytvoří seznam $(y_i, Y_i)_{1 \le i \le q2}$. Pokud v něm najde dvě stejné n bitové hodnoty Y_i a Y_j pro různé i a j, obdrží kolizi pro zprávy x_i a x_j. Pokud A kolizi nenajde, vrací negativní odpověď (kolize nenalezena). Označme p pravděpodobnost, že A kolizi tímto postupem nalezne a P, že nenalezne. Z definice náhodného orákula f vyplývá, že $\{y_i\}_{1 \le i \le q1}$ je množina q_1 náhodných hodnot z množiny $\{0, 1\}^n$. Z definice náhodného orákula g vyplývá, že $\{Y_i\}_{1 \le i \le q2}$ je množina q_2 náhodně vybraných hodnot z množiny $\{0, 1\}^n$. Pravděpodobnost, že ani v jednom z $q_1 + q_2$ kroků nenalezneme kolizi, je

$$P = \prod_{i=1}^{q_1 - 1}\left(1 - \frac{i}{2^n}\right) * \prod_{i=1}^{q_2 - 1}\left(1 - \frac{i}{2^n}\right) \le^{(2)} \prod_{i=1}^{q_1 - 1}\left(e^{-i/2^n}\right) * \prod_{i=1}^{q_2 - 1}\left(e^{-i/2^n}\right) =$$

$$= e^{-q_1(q_1-1)/2^{n+1} - q_2(q_2-1)/2^{n+1}} = e^{(-q(q-2)/4)/2^n}.$$

V úpravě jsme využili faktu (2) z Lemmatu 1 a volby $q_1 = q_2 = q/2$.

Pravděpodobnost, že kolize nastane je

$$p = 1 - P \geq 1 - e^{(-q(q-2)/4)/2^n} \geq^{(1)}$$

$$\geq (1 - e^{-1}) * (q(q-2)/4)/2^n \geq 0.158 * q(q-2)/2^n.$$

V úpravě jsme využili faktu (1) z Lemmatu 1, tj. že $1 - e^{-z} \geq (1-e^{-1})z$ pro všechna $0 \leq z \leq 1$. V roli z vystupuje výraz $z = (q(q-2)/4)/ 2^n$. Protože podle předpokladů věty je vždy $q \leq 2^{n/2}$, je $z \leq 1$ a úprava byla korektní. Tím je důkaz ukončen.

Nyní budeme dokazovat $\mathrm{Adv\_coll\_NMAC[n](q)} \leq 0.5 * q(q-1)/2^n$.

Postačí dokázat platnost odhadu $\mathrm{Adv\_coll\_NMAC[n](A)} \leq 0.5 * q(q-1)/2^n$

pro libovolného útočníka A a $1 < q \leq 2^n + 1$ (pro větší q je pravá strana nerovnosti větší než jedna). Činnost útočníka si můžeme modelovat pomocí jeho dotazů orákulům f a g. Orákulum f si vytváří tabulku T_f záznamů (x, h, y), kde (x, h) je vstup, zvolený útočníkem a y = f(x, h) je odpověď. Orákulum g si vytváří tabulku T_g záznamů (X, Y), kde X je vstup, zvolený útočníkem a Y = g(X) je odpověď. Počet dotazů orákulu f označme q_1 a počet dotazů orákulu g q_2. Máme $q = q_1 + q_2$. Útočník A hledá kolidující zprávy $M \neq M'$, které včetně doplnění mají jeden nebo několik K-bitových bloků. Zprávy mohou mít odlišný počet bloků. Pokud nastane kolize, jsou dvě možnosti:

- **vnitřní kolize:** Kolize nastala před závěrečnou úpravou hašovací funkce. Útočník může v takovém případě příslušné komprese ukončit a přejít k závěrečné úpravě.
- **závěrečná kolize:** Kolize nenastala před závěrečnou úpravou a nastala až v závěrečné úpravě hašovací funkce.

Vnitřní kolize

Uvažujme nejprve vnitřní kolizi. V tomto případě musí při hašování první zprávy v tabulce T_f vzniknout záznam $(x_i, h_{i-1}, h_i = f(x_i, h_{i-1}))$ a při hašování druhé zprávy záznam $(x_j, h_{j-1}, h_j = f(x_j, h_{j-1}))$, kde $i \neq j$ jsou čísla záznamů (nejsou to indexy bloků zpráv), $(x_i, h_{i-1}) \neq (x_j, h_{j-1})$ a $h_i = h_j$, kde x_i je některý K-bitový blok první zprávy a x_j je některý K-bitový blok druhé zprávy. Pro pevné $i = 2, ..., q_1$ označme C_i jev, že existuje j, $1 \leq j < i$ tak, že j-tý záznam a i-tý záznam v tabulce T_f mají stejné hodnoty ve třetí položce, přičemž $(x_i, h_{i-1}) \neq (x_j, h_{j-1})$. Pravděpodobnost jevu C_i označme $\Pr[C_i]$. Protože f je náhodné orákulum, odpověď y_i na i-tý dotaz vybírá náhodně z množiny o 2^n hodnotách. Pravděpodobnost, že vybraná hodnota se rovná některé z třetích položek v tabulce T_f je menší nebo rovna $(i - 1)/2^n$, tj. $\Pr[C_i] \leq (i-1)/2^n$. Označíme-li P_{int} pravděpodobnost, že nastane vnitřní kolize, dostáváme

$$P_{\mathrm{int}} = \Pr[C_2] + \Pr[C_3] + ... + \Pr[C_{q_1}] \leq \sum_{i=2}^{q_1} \frac{i-1}{2^n} \leq q_1(q_1 - 1)/2^{n+1}.$$

Závěrečná kolize

Nyní budeme uvažovat závěrečnou kolizi. Orákulum si vytváří tabulku T_g se záznamy $(X_i, g(X_i))$, $1 \leq i \leq q_2$.Označme P_{fin} pravděpodobnost, že v tabulce T_g nastane závěrečná kolize.
Pro pevné $i = 2, ..., q_2$ označme C_i jev, že existuje $1 \leq j < i$ tak, že j-tý a i-tý záznam v tabulce T_g mají stejné hodnoty v druhé položce, tj. $g(X_i) = g(X_j)$. Pravděpodobnost jevu C_i označme

Pr[C_i]. Protože g je náhodné orákulum, hodnota g(X_i) je vybírána náhodně z množiny všech 2^n hodnot, a proto $\Pr[C_i] \le (i-1)/2^n$. Jestliže existuje závěrečná kolize, pak musel nastat některý z jevů C_i pro $2 \le i \le q_2$. Odtud máme

$$P_{fin} \le \Pr[C_2] + \Pr[C_3] + \ldots + \Pr[C_{q_2}] \le \sum_{i=2}^{q_2} \frac{i-1}{2^n} \le q_2(q_2-1)/2^{n+1}.$$

Celkový odhad

Jestliže útočník nalezne kolizi, musí nastat alespoň jeden z předchozích případů. Odtud dostáváme

$$\mathrm{Adv\_coll\_NMAC[n]}(\mathrm{A}) \le P_{\mathrm{int}} + P_{fin} \le q_1(q_1-1)/2^{n+1} + q_2(q_2-1)/2^{n+1} \le$$

$$\le [(q_1+q_2)^2 - (q_1+q_2)]/2^{n+1} = (q^2-q)/2^{n+1},$$

což jsme měli dokázat.

9. Dodatek 2: Definice speciální blokové šifry DN (Double Net)

Bude doplněn brzy, po schválení jeho publikace.

10. Dodatek 3: Definice hašovací funkce HDN (Hash Double Net)

Bude doplněn brzy, po schválení jeho publikace.