Hašovací funkce nové generace SNMAC[1]

Vlastimil Klíma

nezávislý kryptolog
v.klima@volny.cz,http://cryptography.hyperlink.cz

Abstrakt

V příspěvku se zabýváme koncepcí hašovacích funkcí nové generace (SNMAC), které byly navrženy v říjnu 2006 [Kli06c]. Ukazujeme, že použití klasické blokové šifry v kompresní funkci je systémovou chybou konstrukce a příčinou současných i budoucích problémů hašovacích funkcí. Ukazujeme nutnost jejího nahrazení novým kryptografickým primitivem, speciální blokovou šifrou (SBŠ). Na bázi SBŠ navrhujeme novou generaci hašovacích funkcí SNMAC. Coron a kol. v roce 2005 [CDMP05] dokázali, že konstrukce SNMAC se limitně stávají výpočetně neodlišitelnými od náhodných orákul a Klíma v roce 2006 [Kli06c] předložil důkazy, že SNMAC jsou výpočetně odolné proti nalezení vzoru a kolize. Takové bezpečnostní vlastnosti nemá žádná jiná konstrukce hašovací funkce. SNMAC je navíc obecnou konstrukcí, která umožňuje návrh různých instancí pomocí různých SBŠ. Proto se domníváme, že tento koncept může být kandidátem na hašovací funkce nové generace.

Klíčová slova: Hašovací funkce, HMAC, NMAC, SNMAC, speciální bloková šifra.

1 Úvod

V tomto příspěvku se zabýváme koncepcí hašovacích funkcí nové generace (SNMAC), které byly navrženy v říjnu 2006 [Kli06c].

2 Bezpečná hašovací funkce

Co je hlavním problémem bezpečné hašovací funkce? S nadsázkou řečeno, tím hlavním problémem je, že žádná taková *prakticky využitelná* funkce neexistuje. Například proto, že by měla být bezkolizní, a přitom je jasné, že (má-li být prakticky použitelná) bezkolizní být nemůže. Když bude mít n bitový kód, například 512 bitů, nemohla by hašovat zprávy, které jsou delší než n (512) bitů. Pak to ale není žádná prakticky použitelná hašovací funkce. To samé je s vlastností jednocestnosti. Všechny "důkazy" jednocestnosti všech používaných hašovacích funkcí jsou založeny na *víře*, že lidé nebudou schopni invertovat nějakou složitou funkci. Ba co víc, nejsilnější hašovací funkce současnosti (třída SHA-2), jsou založeny na *víře*, že NSA je navrhla kvalitně. Skutečně, SHA-2 mají punc kvality, i když žádné důkazy bezpečnosti předloženy nebyly[2]. Je to dokonce ještě horší - nejsou známa jejich návrhová kritéria.[3] Přesto mnozí kryptologové *věří*, že tyto funkce jsou dobré. V kvalitu své funkce *věřil* určitě i prof. Rivest z MIT, když navrhoval tehdy dost silně vyhlížející MD5. Za 14 let poté můžeme vytvářet kolize MD5 během sekund na počítači, který si můžeme koupit v supermarketu.[4] Bude to také platit o funkcích SHA-2? Jistě, nikdo nemůže říci, že to nastane ani nenastane. V tom je ta potíž. Chybí *důkazy* bezpečnosti.

Kryptografové se snaží matematické zákony oklamat a takovou funkci navrhnout.[5] Nicméně nám nic jiného nezbývá pokud chceme využívat tak fantastickou myšlenku, jakou je jednocestná funkce pro zcela praktické věci, jakými jsou digitální podpis a podobně. Nový koncept hledají kryptologové na mnoha akademických a

[1] V tomto příspěvku prezentujeme část výsledků projektu NBÚ Bezpečná hašovací funkce (ST20052005017).
[2] I když existují veřejné studie o bezpečnosti SHA-2, neobsahují potřebné důkazy.
[3] Jedna z cest jak najít bezpečnou hašovací funkci, byla vyjít z SHA-2 a udělat takové úpravy, které znemožní současné útoky a ještě přidat nějakou rezervu. Nevýhodou SHA-2 však je, že jejich návrhová kritéria NSA tají. Zcela zásadní otázka pak je, jestli se nějakou úpravou místo zesílení nezapřičiní naopak porušení některých návrhových kritérií a v důsledku toho oslabení výsledné funkce. Proto touto cestou se může odpovědně vydat pouze NSA a nikdo jiný.
[4] Kolize MD5 na běžném PC v průměru za 17 sekund, program a postup viz [Kli06a]
[5] Proto je údělem kryptografů neustále prohrávat :-)

jiných pracovištích na celém světě. Americký standardizační úřad NIST ohlásil plán, jehož obsahem je mnohaletý program na hledání nové hašovací funkce, podobně jako tomu bylo při hledání blokové šifry AES. Zatím se konalo několik mezinárodních kryptologických konferencí k tomuto problému, ale žádný nový hašovací standard na obzoru není. Proč?

Antoine Joux v panelové diskusi na druhé mezinárodní konferenci NIST v srpnu 2006 v USA dobře charakterizoval situaci, když řekl:

> Antoine Joux:
>
> „Nevíme, co děláme a nevíme, co ve skutečnosti chceme…“

a všichni panelisté s ním souhlasili.

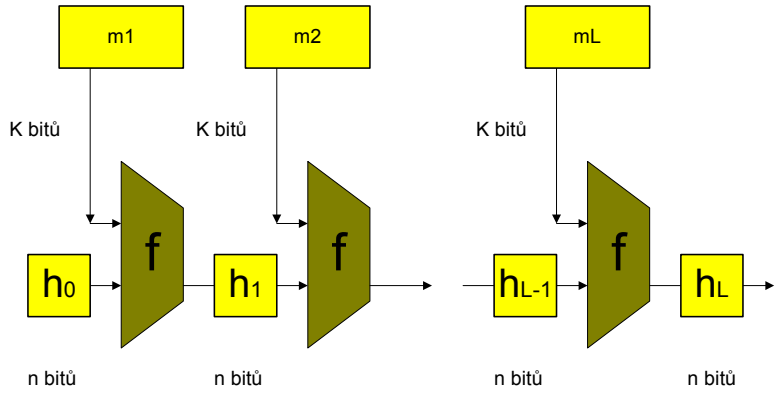
Hlavní příčiny tohoto stavu jsou podle našeho názoru dvě. Za prvé nebyly pojmenovány skutečné příčiny současných útoků na hašovací funkce a za druhé chybí teoretický koncept pro novou generaci hašovacích funkcí. Na oba problémy se snažíme v tomto příspěvku reagovat.

3 Generické útoky

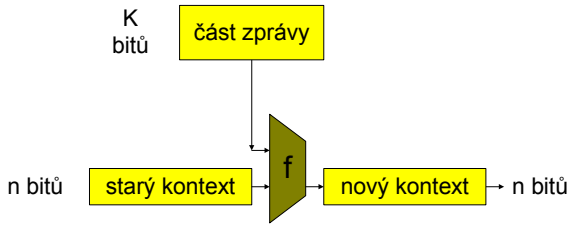
Je známo, že většina používaných iterativních hašovacích funkcí podléhá tzv. útoku prodloužením zprávy [Tsu92], tj. z hodnoty h(M) lze vypočítat h(M || N) pro vhodné N. I když se tím odlišují od náhodného orákula, tato vlastnost byla mnoha hašovacím funkcím dlouho tolerována. V roce 2004 a 2005 byly zjištěny další generické problémy hašovacích funkcí, multikolizní útok Jouxe [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.

4 Iterativní hašovací funkce

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. 1 a 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].

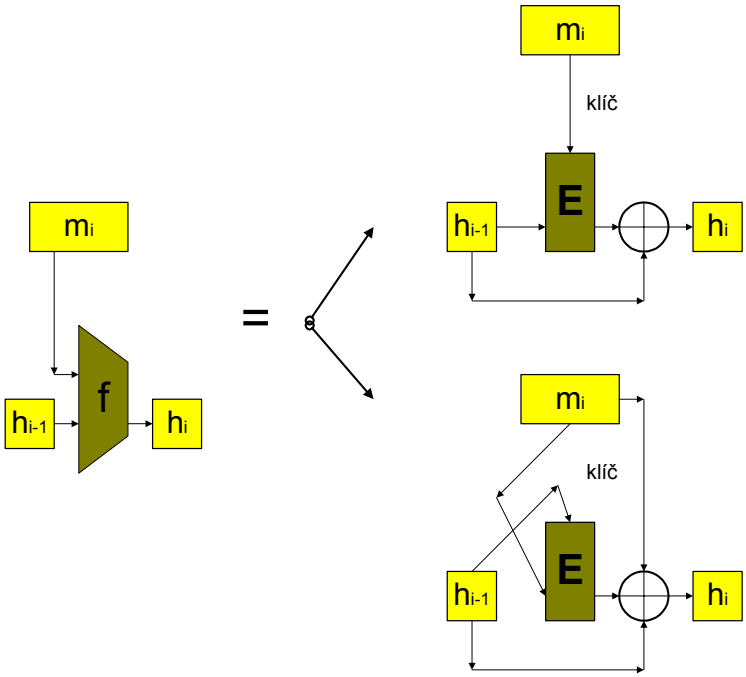


Obr.1: Iterativní hašovací funkce



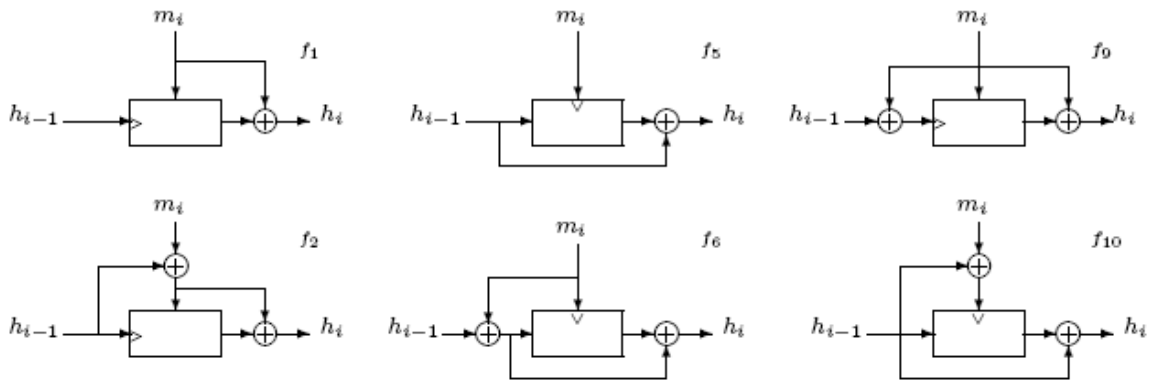
Obr.2:Kompresní funkce

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 a multivzorovým útokům. Vhodnou obranou může být výpočetní složitost. Funkce musíme konstruovat tak, aby tyto útoky vyžadovaly příliš mnoho operací.



Obr.3: Davies-Meyerova a Miyaguchi-Preneelova úprava

Protože hašovací funkce musí zachovávat všechny vlastnosti i při hašování jednoho bloku, můžeme si zkoumání hašovací funkce zjednodušit na jednu aplikaci kompresní funkce, tj. na zkoumání kompresní funkce. Všechny současné hašovací funkce vytváří kompresní funkci z klasické blokové šifry poměrně jednoduchou úpravou. Je to zejména Davies-Meyerova nebo Miyaguchi-Preneelova úprava (obr. 3) nebo některé další navrhované (obr. 4).



Obr.4: Některé úpravy blokové šifry podle [BRS02]

Ve všech těchto případech se vstup do klíče blokové šifry a do otevřeného textu nějak lineárně kombinuje z hodnot kontextu a bloku zprávy a na výstup z blokové šifry se eventuelně také načítá některá nebo obě z těchto hodnot. Tyto úpravy jsou velmi důležité, protože bloková šifra je při pevném klíči permutací, zatímco my potřebujeme, aby kompresní funkce (hašovací funkce) byla náhodným zobrazením a aby nebyla invertibilní. Tu neinvertibilitu potřebujeme proto, aby z hašovacího kódu nešlo určit vzor - hašovanou zprávu. Proto se bloková šifra, která je jinak invertibilní upravuje těmito úpravami.

5 Proč jsou současné hašovací funkce slabé a nejde to spravit?

Současné hašovací funkce jsou slabé proto, že se snaží v kompresní funkci f, viz obr. 2, použít klasickou blokovou šifru. Je zcela jedno, jakým způsobem, ať slabou nebo silnou, jednoduše nebo složitě, jednonásobně nebo vícenásobně. Klasické blokové šifry byly navrhovány s předpokladem, že útočník nezná šifrovací klíč. Jejich cílem bylo pomocí této neznámé hodnoty utajit způsob převodu otevřeného textu na šifrový a naopak. Jsou to tedy stavební prvky, které spoléhají na něco utajeného. V kompresní funkci však nic utajeného není a útočník zná všechny hodnoty, které do ní vstupují. Může je dokonce volit a libovolně s nimi manipulovat, včetně proměnné, která je klíčem klasické blokové šifry. A to je onen základní a podstatný rozpor.

Lze namítnout, že blokovou šifru lze upravit jako na obr. 3 a 4. Jedná se ale o základní a hluboký rozpor. Neprojeví se přímo a hned. Je totiž zakuklen do stavby blokové šifry. Konstruktéři měli u blokové šifry prostě základní předpoklad, že útočník všechno zná, a nezná pouze šifrovací klíč. Kryptografové tuto neznalost "zneužívali" k tomu, aby konstruovali blokové šifry rychlé a bity klíče zpracovávali velmi jednoduše, někdy dokonce vůbec ne. Například u TripleDES, není klíč nijak modifikován a je pouze mnohokrát využíván v čisté podobě. U AES je klíč modifikován pouze velmi slabě oproti funkcím, které jsou použity na modifikaci otevřeného textu. Stavba klasické blokové šifry je proto těžce založena na tom, že velká část jeho vstupu (klíč) je útočníkovi neznámá. A žádná klasická bloková šifra nebyla konstruována ani připravována na situaci, kdy útočník zná šifrovací klíč. To by byl nesmysl. A přesto, u hašovacích funkcí je klíč zcela volně útočníkovi dostupný a může si ho volit, modifikovat a dělat si s ním co chce. Klíčem je většinou zpráva, která se hašuje. Když útočník hledá kolizi nebo vzor, zprávu si může libovolně volit, měnit a tvořit. Manipuluje tedy s klíčem blokové šifry podle potřeby. Na to nebyly klasické blokové šifry připraveny a nikdy proti této možnosti chráněny. Neznalost šifrovacího klíče útočníkem je u klasických blokových šifer zcela základním výchozím bodem. Pokud odstraníme tento předpoklad, už se nejedná o klasickou blokovou šifru, určenou k šifrování. Kdyby někdo chtěl klasickou blokovou šifru použít v hašovací funkci správně, aby byla chráněna i proti útokům ze strany klíče, přeměnil by ji na něco jiného, co má zcela jiné cíle než chránit otevřený text pomocí utajeného klíče. "To něco jiného" je nové kryptografické primitivum, nový základní stavební prvek, který musíme zkoumat a navrhnout. Musíme také říci, co od něj vlastně očekáváme. Nazýváme ho speciální bloková šifra.

Příčinou problémů řady současných hašovacích funkcí je to, že místo kompresní funkce používají stavební prvek, který byl určen pro řešení problému utajení. I přes to by pravděpodobně bylo možné ho upravit a použít ke konstrukci kompresní funkce, ale nelze to současně udělat efektivně, protože ten prvek není primárně konstruován pro použití v kompresní funkci. Protože současné hašovací funkce se snaží být efektivní, dostávají se do rozporu s bezpečností. Další rozpory ukazuje tabulka.

klasická bloková šifra	*kompresní funkce (speciální bloková šifra)*
obsahuje prvek neznámý útočníkovi	útočník zná všechny vstupy kompresní funkce, může s nimi manipulovat
je určena k zakrytí struktury a obsahu otevřeného textu v šifrovém textu na základě tajného prvku, neznámého útočníkovi	je určena k zakrytí struktury a obsahu (nejen části, ale) celého vstupu ve výstupu, neopírá se o žádný utajený prvek
při fixovaném klíči je permutací	je to náhodné zobrazení
je invertibilní	požaduje se neinvertibilita (jednocestnost)
bezkoliznost je nezajímavou vlastností	požaduje se bezkoliznost

Tab.1: rozpory mezi klasickou blokovou šifrou a kompresní funkcí

6 Nové kryptografické primitivum

Podíváme-li se na úlohu kompresní funkce f, chceme, aby byla jednosměrná. Tedy jakákoliv klasická bloková šifra zde není vhodná, protože ta umí dešifrovat. Zde musí být použita funkce, která dešifrovat neumí. Nikoli náhodou se dostáváme do blízkosti kryptografie s veřejným klíčem, neboť ta je založena na jednosměrných funkcích s padacími vrátky - útočník dešifrovat neumí, jen majitel privátního klíče (padacích vrátek). Zde však žádná padací vrátka nesmí existovat, neboť v hašovací funkci nesmí být žádný privátní klíč. Bloková šifra dešifrovat umí, proto, byla různými způsoby modifikována, aby se tato vlastnost narušila a aby se z ní stala jednocestná funkce i při znalosti klíče. Nejpoužívanějšími úpravami jsou Davies-Meyerova a Miyaguchi-Preneelova, viz obr. 3 a 4.

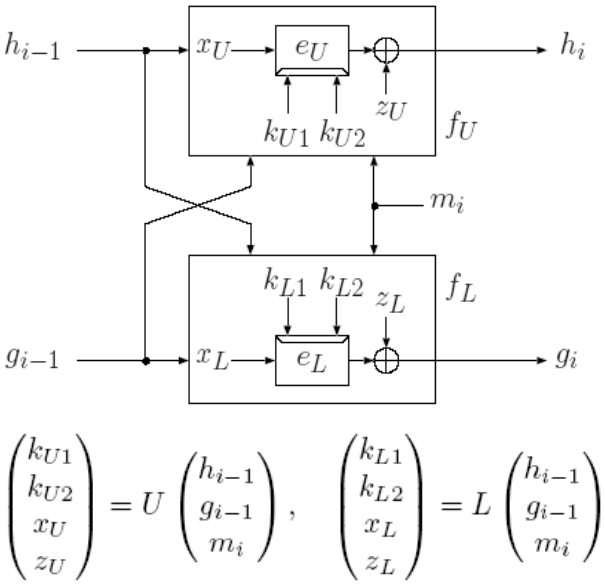
Navrhujeme jinou konstrukci. Vyšli jsme z toho, že znalost klíče nemá poskytnout příliš mnoho informací o otevřeném textu, ideálně žádnou informaci. Jaký zdroj neposkytuje žádnou informaci? Zdroj, který emituje pouze jednu hodnotu, konstantu. Otevřený text musí být tedy konstantní. Pokud bychom navrhli toto primitivum na základě této úvahy, nabízí se konstrukce

$$\mathbf{f: \{0,1\}^K \rightarrow \{0,1\}^n : k \rightarrow E_k(Const_0),}$$

kde E je vhodná (nikoli klasická, ale speciální) bloková šifra a $Const_0$ konstanta. E nemůže být klasickou blokovou šifrou, protože ty nebyly připraveny a konstruovány tak, aby se používaly ve výše uvedené formě. Vidíme, že speciální bloková šifra formálně nic nešifruje. Má ale v názvu "bloková šifra", protože ji budeme konstruovat na pomocí *technologie* (nižších stavebních prvků) blokových šifer.

7 Matematické modely hašovacích funkcí

Víru v bezpečnost hašovacích funkcí (podobně jako u blokových šifer) mají podpořit matematické modely a důkazy jejich vlastností. Tady bylo a dosud je, na rozdíl od blokových šifer, dost velké vakuum. Kromě základního Merkle-Damgardova modelu [Mer89] [Dam89] nebyly dlouho k dispozici žádné jiné modely, které by říkaly něco o kvalitě současných hašovacích funkcí. První dobrou zprávou byla práce Preneela a kol. [PGV03] v roce 2003 (připomeňme, že jeho disertace je základní prací moderní teorie hašovacích funkcí) a následně práce Corona a kol. v roce 2005 [CDMP05]. Hašovací funkce se stavěly s oblibou na bázi blokových šifer jako stavebních bloků. Řada konstrukcí byla studována ve stěžejní práci Blacka a kol. [BRS02], viz obr. 4. Všechny návrhy z [BRS02] však používají příliš jednoduché konstrukce, které dobře propagují diference. Hledala se také robustnější řešení, aby vzniklá konstrukce byla bezpečnější. Příkladem budiž Hiroseho návrh [Hir04] z roku 2004, viz obr. 5. Stále však chyběly přesvědčivé důkazy bezpečnosti.



Obr.5: Konstrukce Hirose [Hir04] : e_U a e_L jsou blokové šifry typu (n,2n), k_{U1}, k_{U2}, x_U, z_U, k_{L1}, k_{L2}, x_L, $z_L \in \{0,1\}^n$ jsou lineární kombinace proměnných h_{i-1}, g_{i-1} a m_i.

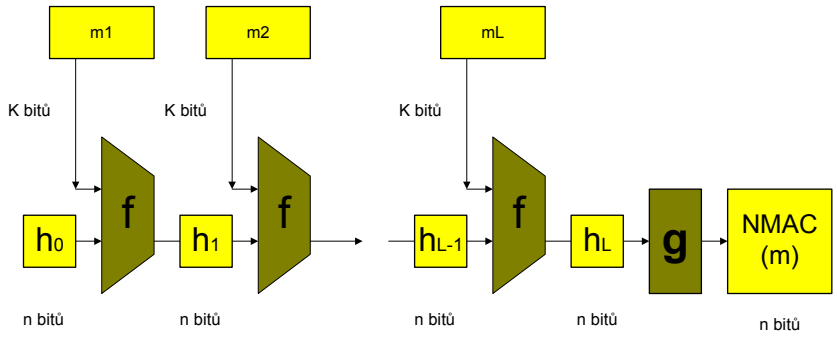
8 Coronovy kvalitativní důkazy o konstrukcích NMAC a HMAC

Zcela zásadní obrat přinesla teoretická práce Corona a kol. [CDMP05] v roce 2005. Ukázali, že určité konstrukce, známé jako NMAC a HMAC, které byly vynalezeny před deseti lety Bellarem a kol. [BCK96], se při zvyšování délek bloku do nekonečna stávají něčím, co je kryptologickým grálem - náhodnými orákuly (limitně se stávají výpočetně neodlišitelnými od náhodných orákul). Přesto jejich příspěvek ještě nebyl zcela doceněn (a nebo se na něm v tichosti pracuje ?)[6].

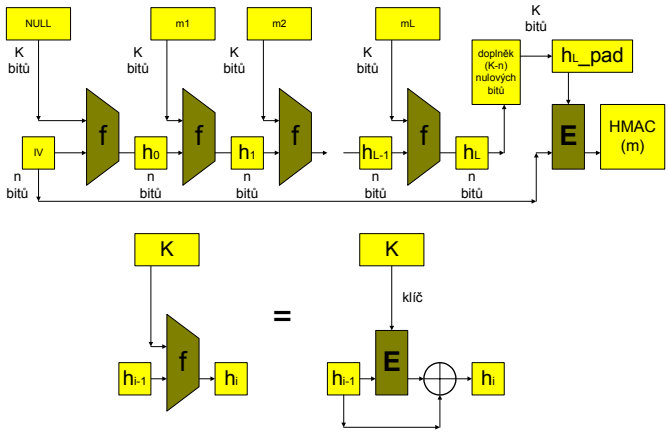
9 Definice NMAC a HMAC

V konstrukci na obr. 6 a 7 je na rozdíl od klasické definice iterativní hašovací funkce použita navíc 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.

[6] NBÚ rozhodl návrh bezpečné hašovací funkce SNMAC zcela zveřejnit k všeobecné diskusi a oponentuře. Nemusí tomu tak být ale u jiných bezpečnostních úřadů ve světě.



Obr.6: Definice hašovací funkce NMAC (viz [BCK96], [CDMP05])



Obr.7: Definice hašovací funkce HMAC (viz [BCK96], [CDMP05])

Pokud použijeme dvě náhodná orákula f a g, dostáváme konstrukci NMAC (viz obr. 6) podle [BCK96], [CDMP05]. Pokud tato orákula konstruujeme pomocí blokové šifry například v Davies-Meyerově formě [MMO85], dostáváme konstrukci, kterou označujeme HMAC (viz obr. 7) podle [BCK96], [CDMP05]. Poznamenáváme, že se jedná formálně o mírně odlišnou definici HMAC, než která je standardizovaná například v RFC2104.

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.

Konstrukce NMAC/HMAC jsou tedy dostatečně odolné proti třetímu generickému útoku - prodloužení zprávy.

10 Naše nové kvantitativní důkazy o konstrukcích NMAC a HMAC

Za základ bezpečné hašovací funkce jsme se rozhodli použít právě Bellare-Coronovy konstrukce NMAC/HMAC. Majíce v ruce důkaz ideální bezpečnosti těchto konstrukcí v nekonečnu, bylo nutné předložit i další důkazy pro konečné rozměry. Víme, že pro konečné rozměry nás u hašovacích funkcí zajímají nejvíce odolnost proti kolizi a proti nalezení vzoru. Zde jsme se nechali inspirovat prací [BRS02], která se zabývala důkazy výše uvedených typů konstrukcí. Přenesli jsme a upravili příslušné metody do prostředí konstrukcí NMAC (které už blokové šifry nepoužívají a používají abstraktnější funkce) a HMAC (používají sice blokové šifry, ale s tzv. závěrečnou úpravou). Obdrželi jsme (matematicky dokázané) zcela konkrétní kvantitativní dolní a horní odhady bezpečnosti. Odhady jsou těsné (dolní a horní mez jsou stejného řádu) a říkají, že k nalezení kolize nebo vzoru u konstrukcí NMAC/HMAC by případný útočník nutně potřeboval řádově tolik operací jako kdyby místo nich byla náhodná orákula!

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

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

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

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

Obr.8: Těsnost odhadů odolnosti (podle vět 1 až 4 [Kli06c]), kde q je počet operací.

Z jedné strany tak máme Coronovy důkazy, které říkají, že kvalitativně se konstrukce NMAC/HMAC blíží ideálu (v nekonečnu) a z druhé strany (pro konečné rozměry) naše důkazy, že jejich odolnost proti nalezení vzoru a kolize je kvantitativně stejná jako u náhodných orákul. Takové vlastnosti nemá prokázány žádná současná hašovací konstrukce.

11 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.

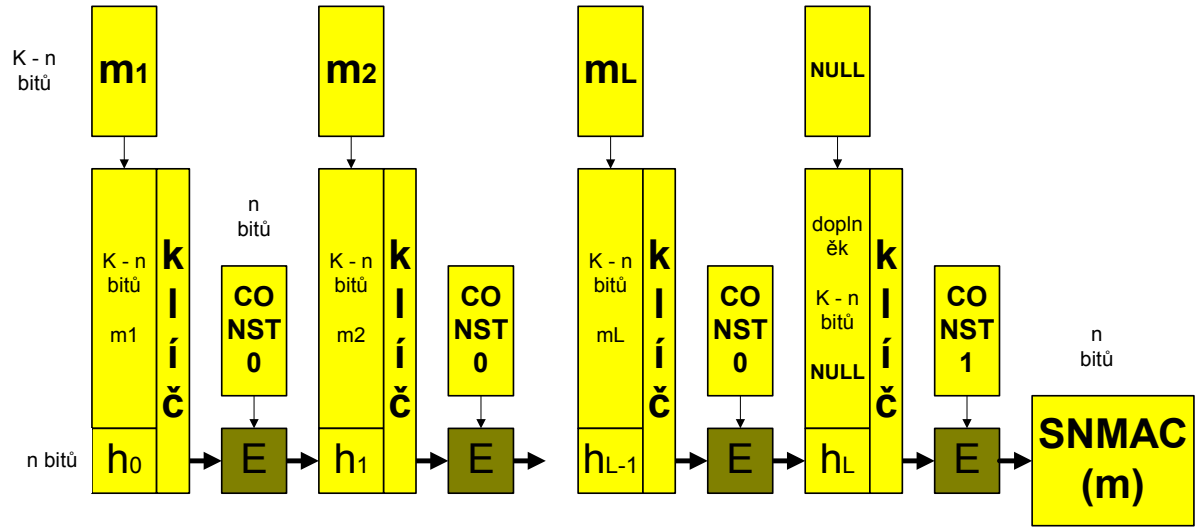
Uvažujme tedy, že v kompresní funkci f, $h_i = f(h_{i-1}, m_i)$ jakýmkoliv způsobem, třeba i několikrát, použijeme blokovou šifru.

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. 9.



Obr. 9: Definice SNMAC, založená na SBŠ a NMAC

Pojem SBŠ je nový a jeho definice se určitě bude ještě vyvíjet.

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 (srv. [BDK07]).

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.

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, $m = m_1 \| \ ... \ \| m_{L-1} \| \ m_L$.

Definujeme (viz obr. 1) 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.

12 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.

13 Závěr

Generické problémy hašovacích funkcí vyvolaly potřebu nového návrhu konceptu hašovacích funkcí. Ukazujeme, že blokové šifry by měly být použity v hašovacích funkcích jiným způsobem než dosud. Nazýváme je speciální blokové šifry (SBŠ) a na jejich bázi navrhujeme novou třídu hašovacích funkcí SNMAC. 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.

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

14 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.

[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.

[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.

[Hir04] S. Hirose: Provably secure double-block-length hash functions in a black-box model, Information Security and Cryptology - ICISC 2004, 7th International Conference, Seoul, Korea, December 2-3, 2004, Lecture Notes in Computer Science, Vol. 3506, pp. 330-342

[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.

[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_CZ.html (in Czech), http://cryptography.hyperlink.cz/SNMAC/SNMAC_EN.html (in English).

[Kli06c] V. Klima: A New Concept of Hash Functions SNMAC Using a Special Block Cipher and NMAC/HMAC Constructions, IACR ePrint archive Report 2006/376 , October, 2006, http://cryptography.hyperlink.cz/SNMAC/SNMAC_EN.pdf (in English), Nový koncept hašovacích funkcí SNMAC s využitím speciální blokové šifry a konstrukcí NMAC/HMAC, http://cryptography.hyperlink.cz/SNMAC/SNMAC_CZ.pdf (in Czech).

[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.

[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

[PGV03] B. Preneel, R. Govaerts, and J. Vandewalle. Hash functions based on block ciphers: A synthetic approach. In Advances in Cryptology – CRYPTO '93, Lecture Notes in Computer Science, pages 368–378. Springer -Verlag, 1994.

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

15 Dodatek 1: Důkazy vět

Viz [Kli06c].

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

Bude zveřejněn po schválení jeho publikace.

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

Bude zveřejněn po schválení jeho publikace.