Crypto-World

Informační sešit GCUCMP

Ročník 7, číslo 4/2005 **15. duben 2005**

4/2005

Připravil: Mgr. Pavel Vondruška
Sešit je přednostně distribuován registrovaným čtenářům.
Starší sešity jsou dostupné na adrese
http://crypto-world.info
(830 registrovaných odběratelů)

Obsah : **str.**

Příloha (PR) : sina.pdf
J.Strelec (Secunet) : SINA - BEZPEČNÁ KOMUNIKAČNÍ INFRASTRUKTURA

A. Co se stalo s hašovacími funkcemi?
aneb přehled událostí z poslední doby, část 2

RNDr. Vlastimil Klíma , http://cryptography.hyperlink.cz , v.klima@volny.cz

Vlastimil Klíma:
"SHA-1 bude do roka prolomena."

Odpověď na dotaz posluchače na semináři
Cryptofest, Praha, 19.3.2005:
Kolik dáváte času SHA-1?

Abstrakt
Z praktického hlediska se loučíme s hašovací funkcí MD5. Z teoretického, a pro mnohé i z praktického hlediska, se loučíme s hašovací funkcí SHA-1. Jako poslední prakticky bezpečné hašovací funkce zůstávají ty ve třídě SHA-2 (funkce SHA-256/384/512/224). Hledá se nový koncept hašovacích funkcí, neboť ani třída SHA-2 nemá ty teoretické vlastnosti, které bychom si u kvalitní hašovací funkce představovali.

III. Blok, týkající se iterativních hašovacích funkcí

V tomto bloku ukážeme, že základ všech moderních hašovacích funkcí - iterativní princip - je teoreticky špatný. Odhalení, která přinesly dvě hlavní práce v minulém roce, ukazují, že iterativní hašovací funkce jsou postaveny na špatném teoretickém základě, který oddaluje tyto funkce od žádoucích náhodných vlastností (od náhodného orákula). Nezbytné pojmy a fakta jsme umístili do dodatku. Jeho text vychází z přednášky [VK2005c], kde naleznete širší výklad k hašovacím funkcím. Nyní uvedeme hodnocení a závěry z obou klíčových prací.

1. Generické problémy iterativních hašovacích funkcí

Generické problémy hašovacích funkcí ukazují dvě práce. První představil Joux [Joux04] na konferenci Crypto v srpnu 2004 a druhý Kelsey-Schneier [KS2004] v listopadu 2004. Obě dvě práce ukazují, že iterativní konstrukce hašovací funkce implikuje značnou odlišnost této funkce od náhodného orákula.

[Joux04b] A. Joux: Multicollisions in iterated hash functions. Application to cascaded constructions. Proceedings of Crypto 2004, LNCS 3152, pages 306-316.
Joux ukazuje, že

1) u iterativních hašovacích funkcí lze docílit **mnohonásobné kolize** mnohem jednodušeji než ve srovnání s náhodným orákulem
2) **kaskádovitá konstrukce** F || G pomocí dvou hašovacích funkcí pozbývá smyslu, neboť očekávaná složitost nalezení kolize není součinem dílčích složitostí, ale spíše součtem

Poznamenejme, že kaskádovitou konstrukci používají někteří architekti k tomu, aby ze dvou slabších funkcí vytvořili silnější nebo aby ze dvou nezávislých hašovacích funkcí s kratším

hašovým kódem vytvořili silnější hašovací funkci s dvojnásobně dlouhým hašovacím kódem, a tedy poskytující neporovnatelně vyšší bezpečnost.

Intuitivně se totiž očekávalo, že složením hašovacího kódu funkce F o délce n_f bitů a hašovacího kódu funkce G o délce kódu n_g bitů vznikne kvalitní hašovací funkce F || G o délce kódu $n_f + n_g$ a složitost nalezení kolize bude tak $2^{(nf+ng)/2}$. Joux ukázal, že místo toho je to mnohem méně, a to $n_g/2 * 2^{nf/2} + 2^{ng/2}$, tedy řádově stále číslo $2^{ng/2}$ nebo $2^{nf/2}$.
Pokud se týká mnohonásobné kolize, jedná se spíše o teoretickou záležitost, ukazující, že hašovací funkce se výrazně odlišuje od náhodného orákula.

[KS2004] John Kelsey, Bruce Schneier: Second Preimages on n-bit Hash Functions for Much Less than 2^n Work, http://eprint.iacr.org/2004/304/, November 15, 2004
Kelsey-Schneierova práce

1) obsahuje výrazně zlepšenou metodu konstrukce multikolizí oproti Jouxovi,
2) **umožňuje konstruovat druhý vzor zprávy** u iterativních hašovacích funkcí se složitostí cca $2*2^{n/2} + 2^{n-k+1}$ pro velmi dlouhé zprávy o délce 2^k blízké $2^{n/2}$.

Konkrétně pro SHA-1 lze ke zprávě o délce 2^{60} bajtů vytvořit druhý vzor se složitostí 2^{106} na rozdíl od teoretické složitosti 2^{160}.

Kelsey-Schneierova práce je velmi významná teoreticky (zejména bod 1), ale má i praktické důsledky (bod 2). Dnes je sice složitost 2^{106} možno považovat za výpočetně nedosažitelnou, ale je to první práce, která umožňuje nalézt druhý vzor zprávy. Připomeneme-li si slova NSA "tyto útoky se mohou pouze zlepšovat", můžeme v budoucnu očekávat snížení této hranice. Pokud by se snížila pod únosnou mez, byl by to největší průlom v oblasti hašovacích funkcí. Zopakujme, že nalezení druhého vzoru zprávy pro hašovací funkci s délkou kódu n bitů má mít teoretickou složitost 2^n, oproti složitosti $2^{n/2}$ pro nalezení kolize. Jsou to tedy neporovnatelně rozdílné úlohy a zde bylo předloženo řešení, které není daleko od hranice bezpečnosti.

Závěr k novým zjištěním kryptoanalýzy iterativních hašovacích funkcí

Řada předních kryptologů se shoduje v tom, že je nutno zahájit práce na veřejné mezinárodní soutěži na nový koncept hašovacích funkcí, neboť iterativní funkce nesplňují požadované bezpečnostní vlastnosti.
Uvedené odhalené vlastnosti jsou teoretického rázu, ale jednoho dne by se mohly projevit zcela prakticky. Proto je nezbytná změna konceptu.

2. NIST doporučuje přechod na SHA-2 do r. 2010

Americký standardizační úřad NIST, který za standardy hašovacích funkcí odpovídá, vydal **25. 8. 2004** prohlášení k tehdejším výsledkům na [NIST05a], z něhož vyjímáme:

- Doporučuje se používat třídu funkcí SHA-2.
- Do roku 2010 se předpokládá opuštění i SHA-1 a přechod na SHA-2.

Po oznámení možnosti nalézt kolizi SHA-1 za 2^{69} operací NIST **23. 2. 2005** svoje dřívější stanovisko ještě více podtrhl [NIST05b].

3. Třída hašovacích funkcí SHA-2 (SHA-256, 384, 512 a 224)

Z důvodu zvýšení odolnosti vůči kolizím je od 1. února 2003 k dispozici nová trojice hašovacích funkcí SHA-256, SHA-384 a SHA-512 [SHA-2] a od února 2004 SHA-224 (dodatek [SHA-2]). Tyto funkce přichází se zvýšením délky hašového kódu na 256, 384 a 512 bitů (SHA-224 má 224 bitový hašový kód), což odpovídá složitosti 2^{128}, 2^{192} a 2^{256} pro nalezení kolizí narozeninovým paradoxem. To je jednak už dost vysoká složitost a také to odpovídá složitosti útoku hrubou silou na tři délky klíčů, které nabízí standard AES.
Pokud se týká konstrukce nových funkcí, jsou velmi podobné SHA-1 a používají stejné principy, pracují však se složitějšími funkcemi a širšími vstupy. Podrobnosti lze nalézt v uvedených standardech. Jejich cílem bylo poskytnout větší odolnost proti kolizi a nabídnout odpovídající bezpečnost jako klíče pro AES.

Tyto funkce jsou iterativního charakteru, takže mají teoretické nedostatky, zmíněné výše. Proto by bylo vhodné přejít na jiný koncept konstrukce.

4. Hodnotící práce

[HOSCH05] P. Hoffman, B. Schneier: **Attacks on Cryptographic Hashes in Internet Protocols**, Internet-Draft, March 25, 2005, http://www.ietf.org/internet-drafts/draft-hoffman-hash-attacks-00.txt
Dokument *sumarizuje vše, co je známo o útocích* na hašovací funkce a týká se internetových protokolů, takže jeho platnost jsou ve skutečnosti všeobecné. Zároveň konstatuje, že existuje všeobecná neshoda v tom, co z toho vyplývá a jak reagovat. Dokonce se konstatuje, že i autoři tohoto internetového dokumentu mají odlišný názor na to, jak reagovat na současné útoky. Nicméně se shodují v tom, že migrace na SHA-256 není rozhodně na škodu plus že aplikace by měly být připraveny na použití hašovacích funkcí s delším kódem. Pokud tyto vlastnosti nemají, měly by být brzo opraveny. Bruce Schneier k tomu dodává úsloví, které se traduje v NSA: *Útoky se pouze vylepšují, nikdy nezhoršují*. A dále říká: *současné kolize MD5 se dají najít na jednom počítači, útoky proti SHA-1 jsou zatím za horizontem, ale budou se pouze zlepšovat. Délka 256 bitů SHA-256 nám dá mnohem větší bezpečnostní polštář v případě objevení eventuelních dalších útoků. Během několika dalších let by pak kryptografická komunita měla dát zlepšené návrhy konstrukce hašovacích funkcí.*

Poznámka. Schneier zde cituje práci [VK2005a], která má však už aktualizaci i s popisem metody hledání kolizí [VK2005b].

[EU05] Katholieke Universiteit Leuven: **Recent Collision Attacks on Hash Functions**, ECRYPT Position Paper, kontraktor (autor dokumentu): Katholieke Universiteit Leuven, 17. February 2005, Revision 1.1
http://www.ecrypt.eu.org/documents/STVL-ERICS-2-HASH_STMT-1.1.pdf
Stručný závěr: Kde je to možné, přejít na SHA-2 nebo Whirpool, funkce MD5 a SHA-1 používat pouze tam, kde nevadí narušení vlastnosti bezkoliznosti, tedy nepoužívat v digitálních podpisech. V kódech HMAC je možné MD5 a SHA-1 používat. Jedná se o shrnující práci projektu EU ECRYPT.

[LE05] Arjen K. Lenstra: **Further progress in hashing cryptanalysis**,
February 26, 2005, http://cm.bell-labs.com/who/akl/hash.pdf
Stručný závěr: Nepoužívat MD5 pro potřeby certifikátů. Nové produkty by měly používat SHA-2. Existující systémy, používající MD5 nebo SHA-1 jsou ohroženy, pokud se spoléhají na vlastnost bezkoliznosti. Nejsou zatím ohroženy ty systémy, kde by jejich narušení požadovalo konstrukci druhého vzoru.

[VK2005c] Vlastimil Klíma: **Hašovací funkce, principy, příklady a kolize**, přednáška na semináři Cryptofest, http://www.cryptofest.cz/, Praha, 19.3. 2005, http://cryptography.hyperlink.cz/2005/cryptofest_2005.htm.
Stručný závěr a doporučení z této práce jsou uvedeny dále.

5. Které techniky jsou a které nejsou bezpečné

- Prolomené hašovací funkce by se neměly používat tam, kde se jedná o nepopiratelnost, tedy u digitálních podpisů. Neměly by se používat tam, kde je důležitá vlastnost bezkoliznosti, kde útočník může využít náhodné kolize.
- **Klíčované hašové autentizační kódy zpráv HMAC ani pseudonáhodné funkce PRF a pseudonáhodné generátory PRNG, které používají hašovací funkce jako nástroje, zatím nejsou současnou kryptoanalýzou dotčeny**. (Pojmy PRNG, PRF a HMAC viz například kapitoly 9 - 11 [VK2005c]).
- Je tu ale možné riziko pramenící z toho, že máme jen velmi málo informací o technikách prolomení současných hašovacích funkcí a že lze v této oblasti očekávat pokrok. To je hrozba, kterou si každý musí ohodnotit.
- Dále víme, že iterativní konstrukce hašovací funkce vede k rozporu s vlastnostmi náhodného orákula. To může časem také přinést nová odhalení vlivu této konstrukce na kvalitu PRNG s těmito hašovacími funkcemi.

6. Doporučení

- Je vhodné provést revizi všech aplikací, kde jsou použity hašovací funkce MD4, MD5, SHA-0, RIPEMD a HAVAL-128.
- Je-li některá z těchto funkcí použita pro účely digitálních podpisů (s klasickým účelem zajištění nepopiratelnosti), je nutno tuto funkci nahradit.
- U funkce SHA-1 je nutné ji nahradit nebo zvážit riziko jejího ponechání v každé aplikaci. Jde zejména o možnost vzniku škody "zpětně", tj. argumentací v budoucnu, kdy by byla SHA-1 prolomena, že mohla být prolomena již teď, tj. v minulosti.
- Podle okolností provést náhradu za některou z funkcí SHA-2, které jsou zatím považovány za bezpečné (SHA-256, SHA-384 nebo SHA-512, nejlépe SHA-512 [SHA-1,2]).
- Je-li některá z prolomených hašovacích funkcí použita pro účely HMAC, PRF nebo PRNG, individuálně posoudit, zda je toto užití bezpečné nebo ne.

7. Dodatek

Tento dodatek obsahuje některé pojmy a informace, tvořící základ pro závěry uvedené v předchozí části. Text vychází z přednášky [VK2005c], kde naleznete širší výklad k hašovacím funkcím. Na stránce,věnované kolizím hašovacích funkcí http://cryptography.hyperlink.cz/2004/kolize_hash.htm naleznete také další literaturu v češtině i v angličtině, vztahující se ke konkrétním hašovacím funkcím a technikám.

Orákulum a náhodné orákulum

Orákulum nazýváme libovolný stroj (stroj "podivuhodných vlastností"), který na základě vstupu odpovídá nějakým výstupem. Má pouze vlastnost, že na tentýž vstup odpovídá tímtéž výstupem. Náhodné orákulum je orákulum, které na nový vstup odpovídá náhodným výběrem výstupu z množiny možných výstupů.

Hašovací funkce jako náhodné orákulum

Z hlediska bezpečnosti bychom byli rádi, kdyby se hašovací funkce chovala jako náhodné orákulum. Odtud se odvozují bezpečnostní vlastnosti.

Bezpečnost z hlediska nalezení vzoru, prolomení hašovací funkce poprvé

Pokud se bude hašovací funkce f: $\{0,1\}^D \rightarrow \{0,1\}^n$ chovat jako náhodné orákulum, bude složitost nalezení vzoru k danému hašovacímu kódu rovna 2^n.
Pokud je nalezena cesta, jak vzory nalézat jednodušeji, hovoříme o prolomení hašovací funkce.

Složitost nalezení kolize

Jestliže kolize zákonitě existují, položme si otázku, jak velká musí být množina náhodných zpráv, aby v ní s nezanedbatelnou pravděpodobností existovaly dvě různé zprávy se stejnou haší. Narozeninový paradox říká, že pro n-bitovou hašovací funkci nastává kolize s cca 50% pravděpodobností v množině $2^{n/2}$ zpráv, namísto očekávaných $1/2 * 2^n$. Například pro 160bitový hašový kód bychom očekávali $1/2 * 2^{160}$ zpráv, paradoxně je to pouhých 2^{80} zpráv.

Tvrzení (narozeninový paradox)

Mějme množinu M m různých koulí a proveďme výběr k koulí po jedné s vracením do množiny M. Potom pravděpodobnost, že vybereme některou kouli dvakrát nebo vícekrát je $P(m, k) = 1 - m(m-1)...(m - k + 1)/m^k$. Pro $k = O(m^{1/2})$ a m velké je $P(m, k) \approx 1 - \exp(-k^2/2m)$.

Důsledek

Pro m velké se ve výběru $k = (2m * \ln_e 2)^{1/2} \approx m^{1/2}$ prvků z M s cca 50% pravděpodobností naleznou dva prvky shodné.

Paradoxnost.

Běžně by člověk uvažoval následovně. Máme množinu m prvků, vezmeme si jeden prvek a hledáme k němu druhý. Abychom dostali pravděpodobnost 1/2, musíme vytahat asi polovinu množiny M, tj. m/2 prvků. Místo toho ale postačí odmocnina z m prvků.

Máme P(365, 23) = 0.507. Pro čísla m = 365 a k = 23 interpretujeme tvrzení tak, že skupina 23 náhodně vybraných lidí postačí k tomu, aby se mezi nimi s cca 50%

pravděpodobností našla dvojice, slavící narozeniny tentýž den. U skupiny 30 lidí je pravděpodobnost už P(365, 30)= 0.706.

Tvrzení se zdá paradoxní protože, ač je vyřčeno jinak, obvykle ho vnímáme ve smyslu "kolik lidí je potřeba, aby se k danému člověku našel jiný, slavící narozeniny ve stejný den". V této podbízející se interpretaci hledáme jedny konkrétní narozeniny, nikoli "jakékoliv shodné" narozeniny. Oba přístupy odráží přesně rozdíl mezi kolizí prvního řádu (libovolní dva lidé) a druhého řádu (nalezení druhého člověka k danému).

Multikolize

Multikolizí (r-násobnou kolizí, r-cestnou kolizí) nazýváme r-tici různých zpráv, vedoucích na stejnou haš.

r-násobná kolize u náhodného orákula

K tomu, abychom mezi odpověďmi náhodného orákula na N různých dotazů nalezli jednu odpověď r krát (r-násobnou kolizi), postačí s dostatečnou nenulovou pravděpodobností $N = 2^{n*(r-1)/r}$ dotazů, což je pro větší r přibližně 2^n. Pro r = 2 dostáváme známý narozeninový paradox a složitost $2^{n/2}$. Pojmem r-násobné kolize se poprvé zabýval Merkle na konferenci Crypto 1989.

Bezpečnost z hlediska nalezení kolize, prolomení hašovací funkce podruhé

Pokud se hašovací funkce $f: \{0,1\}^D \rightarrow \{0,1\}^n$ bude chovat jako náhodné orákulum, bude složitost nalezení kolize rovna přibližně $2^{n/2}$ a složitost nalezení r-násobné multikolize přibližně $2^{n*(r-1)/r}$.

Pokud je nalezena cesta, jak kolize nalézat jednodušeji, hovoříme o prolomení hašovací funkce.

Prakticky používané hašovací funkce nejsou prokazatelně bezpečné

I když uvidíme, že vytváření hašovacích kódů je opravdu neskutečně složité, nalezení kolizí je přesto pouze otázkou intelektuální výzvy, neboť u prakticky používaných hašovacích funkcí není prokázána výpočetní složitost nalezení kolize nebo druhého vzoru. Jejich bezpečnost tak u obou vlastností (jednosměrnost, bezkoliznost) závisí pouze na stavu vědy v oblasti kryptografie a kryptoanalýzy.

Prolomení některých kryptografických technik je proto přirozeným a průvodním jevem rozvoje poznání v této oblasti.

Když je nalezena kolize

Hašovací funkce, u níž byla nalezena kolize, ztrácí generálně smysl, neboť hypotéza o tom, že se chová jako náhodné orákulum byla vyvrácena. Zejména by neměla být používána k digitálním podpisům, neboť tam kolize znamená, že je možné předložit dvě různé zprávy s tímtéž platným digitálním podpisem, platným pro obě zprávy. Existují ale techniky, kde nejsou využity všechny vlastnosti hašovací funkce a kde porušení bezkoliznosti (nebo částečné porušení bezkoliznosti) nevadí (PRNG, PRF, HMAC).

Generické problémy iterativních hašovacích funkcí

Generické problémy hašovacích funkcí ukazují dvě práce. První představil Joux [Joux04] na konferenci Crypto v srpnu 2004 a druhý Kelsey-Schneier [KS2004] v listopadu 2004. Obě

dvě práce ukazují, že iterativní konstrukce hašovací funkce implikuje značnou odlišnost této funkce od náhodného orákula.

[Joux04b] A. Joux: Multicollisions in iterated hash functions. Application to cascaded constructions. Proceedings of Crypto 2004, LNCS 3152, pages 306-316.
[KS2004] John Kelsey, Bruce Schneier: Second Preimages on n-bit Hash Functions for Much Less than 2^n Work, http://eprint.iacr.org/2004/304/, November 15, 2004

Joux ukazuje, že

3) u iterativních hašovacích funkcí lze docílit **mnohonásobné kolize** mnohem jednodušeji než ve srovnání s náhodným orákulem
4) **kaskádovitá konstrukce** F || G pomocí dvou hašovacích funkcí pozbývá smyslu, neboť očekávaná složitost nalezení kolize není součinem dílčích složitostí, ale spíše součtem

Kelsey-Schneierova práce

3) obsahuje výrazně zlepšenou metodu konstrukce multikolizí oproti Jouxovi,
4) **umožňuje konstruovat druhý vzor zprávy** u iterativních hašovacích funkcí se složitostí cca $2*2^{n/2} + 2^{n-k+1}$ pro velmi dlouhé zprávy o délce 2^k blízké $2^{n/2}$.

Konkrétně pro SHA1 lze ke zprávě o délce 2^{60} bajtů vytvořit druhý vzor se složitostí 2^{106} na rozdíl od teoretické složitosti 2^{160}.

r-násobná kolize pro iterativní hašovací funkce lze docílit s nižší složitostí

Joux [Joux04b] ukázal, že u iterativních hašovacích funkcí lze r-násobnou kolizi najít se složitostí $\ln_2 r * 2^{n/2}$ namísto $2^{n*(r-1)/r}$.

Postup. Vyjdeme ze standardní hodnoty IV, H_0 = IV, se složitostí S(F) najdeme kolizi hašovací funkce F s inicializační hodnotou H_0 (zprávy $M_{1,1}$ a $M_{1,2}$). Výslednou haš označme H_1. Se složitostí S(F) nalezneme kolizi F s inicializační hodnotou H_1 (zprávy $M_{2,1}$ a $M_{2,2}$), výslednou haš označíme H_2. Takto uděláme N kroků pro $N = \ln_2 r$. Nyní můžeme sestavit 2^N = r zpráv, majících tutéž haš H_N, a to tak, že z každé dvojice bloků $M_{i,1}$ a $M_{i,2}$ vybereme vždy jednu z nich. Dostaneme tak 2^N zpráv, které prochází stejnými hašovacími mezivýsledky a končí stejným hašovacím kódem H_N.

Kaskádovitá konstrukce pozbývá smyslu

Druhou vlastností, kterou Joux [Joux04b] odhalil, je že složení hašovacích funkcí F a G (kaskáda), F||G (|| označuje zřetězení) neposkytuje intuitivně předpokládanou bezpečnost, ale mnohem nižší. Předpokládalo se, že složitost S(F||G) nalezení kolize hašovacího kódu F(x)||G(x) bude rovna součinu složitostí nalezení kolizí dílčích hašovacích kódů, tj. S(F||G) = S(F) * S(G). Joux ukázal, že je to jen o něco více než S(F) + S(G), přičemž postačí, aby pouze F byla iterativní hašovací funkce, zatímco G může být i náhodné orákulum. Stručně řečeno kaskádovitá konstrukce pozbývá smyslu, protože výsledný kód je přibližně pouze tak složitý jako silnější z dílčích hašovacích funkcí. Tyto dvě vlastnosti přímo neohrožují žádné prakticky používané schéma, ale ukazují, že iterativní konstrukce není ideální, neboť oddaluje takové hašovací funkce od náhodného orákula.

Postup.

- Nechť F je iterativní hašovací funkce s délkou hašovacího kódu $n_f \leq n_g$.
- Potom se složitostí $n_g/2 * S(F)$ vytvoříme $n_g/2$ návazných kolizí funkce F (postup stejný jako použil Joux), které dávají $2^{ng/2}$ - násobnou multikolizi vzhledem k F.
- Mezi těmito $2^{ng/2}$ zprávami nalezneme jednu kolizi vzhledem ke G.
- Máme tedy dvě zprávy, které mají stejný hašový kód vzhledem k F i G, tj. k F||G.

Složitost je $n_g/2 * S(F) + 2^{ng/2}$ (druhý sčítanec je počet hašování G), tedy $n_g/2 * S(F) + S(G) \approx S(F) + S(G)$.
Intuitivně se očekávalo, že složením kvalitní hašovací funkce F o délce kódu n_f a funkce G o délce kódu n_g vznikne kvalitní hašovací funkce o délce kódu $n_f + n_g$ a složitosti nalezení kolize bude $2^{(nf+ng)/2}$. Místo toho je to mnohem méně, $n_g/2 * 2^{nf/2} + 2^{ng/2}$.

Nalezení druhého vzoru u dlouhých zpráv snadněji než se složitostí 2^n

V práci [KS2004] se tato vlastnost ukazuje pro dlouhé zprávy, o délce blízké $2^{n/2}$ bloků. Postup (zkrácený postup s využitím pevných bodů).

- Nechť zpráva M má délku 2^k bloků.
- Vytvoříme seznam průběžných kontextů K_t při hašování zprávy $M = m_1, m_2, ..., m_t, ...$ Je jich 2^k.
- Volíme náhodně $2^{n/2}$ bloků M_i , které dávají seznam $2^{n/2}$ haší $h_i = h(H_0, M_i)$.
- Volíme náhodně $2^{n/2}$ bloků N_j a z N_j určíme pevný bod $H_j = f(H_j , N_j)$, využijeme k tomu Davies-Meyerovy konstrukce.
- Nalezneme kolizi mezi seznamy $\{H_j\}$ a $\{h_i\}$, tj. i* a j* tak, že $h_{i*} = H_{j*}$.
- Volíme náhodně $2^{n - k}$ bloků $Mlink_l$, l = 1, 2, ..., $2^{n - k}$, které dávají seznam $2^{n - k}$ haší $hlink_l = h(H_{j*}, Mlink_l)$.
- Nalezneme kolizi mezi seznamy $\{hlink_l\}$ a $\{K_t\}$, tj. l* a t* tak, že $hlink_{l*} = K_{t*}$.
- Zpráva (M_{i*}, N_{j*}, $Mlink_{l*}$) a prvních *i* bloků zprávy M dávají stejný hašovací kontext K_{t*} .
- Tyto zprávy mají různou délku, ale zprávu (M_{i*}, N_{j*}, $Mlink_{l*}$) doplníme o potřebný počet bloků na i bloků pomocí pevného bodu, jako (M_{i*}, N_{j*}, N_{j*},..., N_{j*}, $Mlink_{l*}$). Za obě zprávy pak připojíme zbytek zprávy M a dostaneme druhý vzor zprávy M.

Složitost je $2^{n/2}$ (seznam M_i) + $2^{n/2}$ (seznam N_j) + 2^{n-k} (seznam $Mlink_l$) + 2^k (seznam K_t) = $2^{n/2+1} + 2^{n-k} + 2^k \approx 2^{n/2+1} + 2^{n-k+1}$. To je mnohem méně než 2^n.

8. Literatura

[ARCHIV] Archiv autora obsahující články o kryptologii a bezpečnosti, http://cryptography.hyperlink.cz/
[BC04a] Biham, Eli, Chen, Rafi: Near Collisions of SHA-0, CRYPTO 2004
http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2004/CS/CS-2004-09.ps.gz

[BC04b] Eli Biham, Rafi Chen: New results on SHA-0 and SHA-1, CRYPTO 2004 Rump Session

[BoBo93] B. den Boer and A. Bosselaers. Collisions for the compression function of MD5. In Advances in Cryptology, Eurocrypt '93, pages 293-304, Springer-Verlag, 1994.

[D96a] H. Dobbertin, Cryptanalysis of MD4, Fast Software Encryption 1996, LNCS, Vol. 1039, Springer-Verlag, 1996, pp. 53 - 69

[DK2004] Dan Kaminsky: MD5 To Be Considered Harmful Someday, *Cryptology ePrint Archive*, Report 2004/357, http://eprint.iacr.org/2004/357, 6 December 2004

[DO96eu] H. Dobbertin. Cryptanalysis of MD5 Compress. Presented at the rump session of Eurocrypt '96, May 14, 1996.

[DO96cb] H. Dobbertin. The Status of MD5 after a Recent Attack. CryptoBytes, 2(2): 1-6, 1996.

[HAVAL] Y. Zheng, J. Pieprzyk, J. Seberry, HAVAL - A One-way Hashing Algorithm with Variable Length of Output, Auscrypt 92

[HMAC] FIPS PUB 198, The Keyed-Hash Message Authentication Code (HMAC), NIST, US Department of Commerce, Washington D. C., March 6, 2002, http://csrc.nist.gov/CryptoToolkit/tkhash.html, resp. RFC 2104, http://www.rfc-editor.org/

[HPR04] Philip Hawkes, Michael Paddon, Gregory G. Rose: Musings on the Wang et al. MD5 Collision, *Cryptology ePrint Archive*, Report 2004/264, 13 October 2004, http://eprint.iacr.org/2004/264.pdf

[Joux04a] Antoine Joux: Collisions in SHA-0, CRYPTO 2004 Rump Session

[Joux04b] A. Joux: Multicollisions in iterated hash functions. Application to cascaded constructions. Proceedings of Crypto 2004, LNCS 3152, pages 306-316.

[KS2004] John Kelsey, Bruce Schneier: Second Preimages on n-bit Hash Functions for Much Less than 2^n Work, http://eprint.iacr.org/2004/304/, November 15, 2004

[LWW05a] Arjen Lenstra, Xiaoyun Wang and Benne de Weger: Colliding X.509 Certificates, *Cryptology ePrint Archive*, Report 2005/067, http://eprint.iacr.org/2005/067

[LWW05b] Arjen Lenstra, Xiaoyun Wang and Benne de Weger: Colliding X.509 Certificates based on SHA1-collisions, http://www.win.tue.nl/~bdeweger/CollidingCertificates/index.html

[MD245] MD2, MD4, MD5 - RFC 1319, 1320, 1321, http://www.rfc-editor.org/

[NIST05b] NIST Brief Comments on Recent Cryptanalytic Attacks on SHA-1 http://csrc.nist.gov/news-highlights/NIST-Brief-Comments-on-SHA1-attack.pdf

[NIST05a] NIST Brief Comments on Recent Cryptanalytic Attacks on Secure Hashing Functions and the Continued Security Provided by SHA-1, http://csrc.ncsl.nist.gov/hash_standards_comments.pdf,

[OOW94] P. van Oorschot and M. Wiener. Parallel collision search with application to hash functions and discrete logarithms. In Proceedings of 2nd ACM Conference on Computer and Communication Security, pages 210-218, ACM Press, 1994.

[OM2004] Ondrej Mikle: Practical Attacks on Digital Signatures Using MD5 Message Digest, *Cryptology ePrint Archive*, Report 2004/356, http://eprint.iacr.org/2004/356, 2nd December 2004, http://cryptography.hyperlink.cz/2004/collisions.htm

[PKCS2] *PKCS#5 v2.0: Password-Based Cryptography Standard*, RSA Labs, March 25, 1999

[RIPEMD-160] H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160: A Strengthened Version of RIPEMD," Fast Software Encryption, LNCS 1039, D.Gollmann, Ed., Springer-Verlag, 1996, pp. 71-82

[SHA-0] FIPS 180 (superseded by FIPS 180-1 and FIPS 180-2), Secure hash standard (SHS), NIST, US Department of Commerce, Washington D. C., May 1993

[SHA-1] FIPS 180-1 (superseded by FIPS 180-2), Secure hash standard (SHS), NIST, US Department of Commerce, Washington D. C., April 1995

[SHA-2] FIPS 180-2, Secure Hash Standard (SHS), NIST, US Department of Commerce, Washington D. C., August 2002 (change notice: February 2004), http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf, platný standard, obsahuje definice SHA-1, SHA-224, SHA-256, SHA-384 a SHA-512

[VK2005a] Vlastimil Klima: Finding MD5 Collisions – a Toy For a Notebook, *Cryptology ePrint Archive*, Report 2005/075, March 5, 2005, http://eprint.iacr.org/2005/075, v češtině " Nalézání kolizí MD5 - hračka pro notebook", http://cryptography.hyperlink.cz/md5/MD5_kolize.pdf.

[VK2005b] Vlastimil Klima: Finding MD5 Collisions on a Notebook PC Using Multi-message Modifications, March 31, 2005, *Cryptology ePrint Archive*, Report 2005/112, http://eprint.iacr.org/2005/102, v češtině " Nalézání kolizí MD5 na notebooku pomocí mnohonásobných modifikací zprávy", http://cryptography.hyperlink.cz/md5/Vlastimil_Klima_MD5_kolize.pdf .

[VK2005c] Vlastimil Klíma: Hašovací funkce, principy, příklady a kolize, přednáška na semináři Cryptofest, http://www.cryptofest.cz/, Praha, 19.3. 2005, on line na http://cryptography.hyperlink.cz/2005/cryptofest_2005.htm.

[WFLY04] X. Wang, D. Feng, X. Lai, H. Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", rump session, CRYPTO 2004, *Cryptology ePrint Archive*, Report 2004/199, http://eprint.iacr.org/2004/199

[WY2005] Xiaoyun Wang and Hongbo Yu: How to Break MD5 and Other Hash Functions, http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf.

[WYY05] Wang X., Yin L., Yu H.: Collision Search Attacks on SHA1, February 13, 2005, http://theory.lcs.mit.edu/~yiqun/shanote.pdf